Towards Operational Measures of Computer Security

Bev Littlewood
Sarah Brocklehurst
Norman Fenton
Peter Mellor
Stella Page
David Wright

Centre for Software Reliability
City University
Northampton Square
London EC1V 0HB

John Dobson

Computing Laboratory
University of Newcastle-upon-Tyne
Newcastle-upon-Tyne NE1 7RU

John McDermid

Dept of Computer Science
University of York
Heslington
York Y01 5DD

Dieter Gollmann

Dept of Computer Science
Royal Holloway and Bedford New College
Egham Hill
Egham TW20 0EX

Abstract

Ideally, a measure of the security of a system should capture quantitatively the intuitive notion of 'the ability of the system to resist attack'. That is, it should be *operational*, reflecting the degree to which the system can be expected to remain free of security breaches under particular conditions of operation (including attack). Instead, current security *levels* at best merely reflect the extensiveness of safeguards introduced during the design and development of a system. Whilst we might expect a system developed to a higher level than another to exhibit 'more secure behaviour' in operation, this cannot be guaranteed; more particularly, we cannot infer what the actual security behaviour will be from knowledge of such a level. In the paper we discuss similarities between reliability and security with the intention of working towards measures of 'operational security' similar to those that we have for reliability of systems. Very informally, these measures could involve expressions such as the rate of occurrence of security breaches (cf rate of occurrence of failures in reliability), or the probability that a specified 'mission' can be accomplished without a security breach (cf reliability function). This new approach is based on the analogy between *system failure* and *security breach*. A number of other analogies to support this view are introduced. We examine this duality critically, and have identified a number of important open questions that need to be answered before this quantitative approach can be taken further. The work described here is therefore somewhat tentative, and one of our major intentions is to invite discussion about the plausibility and feasibility of this new approach.

1 Introduction

Current approaches to measuring and predicting system reliability [Laprie 1989; Littlewood 1989] are based on the definition of reliability in terms of the probability of failure-free operation for a specified length of time. The advantage of this operational approach is that it allows us to obtain measures of the actual *behaviour* of a device, or of a computer program, as seen by a user, rather than merely of some static properties. Thus users are likely to be more interested in knowing the reliability of their system, expressed for example as a rate of occurrence of failures (ROCOF), than in knowing that it possesses certain structural properties or measures, or that it was developed under a particular regime. These static attributes and measures of the system, and its development process, undoubtedly do influence the user-perceived operational reliability, but they are not sufficient in themselves to determine this reliability.

The present position in security seems to be that current ‘measures’, or rankings, of security of systems are merely analogous to the static attributes and measures discussed above. The Orange Book [NCSC 1985] levels, for example, are concerned with those factors which are likely to *influence* ‘operational security’, and may very well be beneficial in producing secure systems, but they do not facilitate quantitative *evaluation* of the actual achieved operational security. Indeed, it is not possible even to know that a particular system that has reached a higher level in this scheme than another system is, in some real operational sense, ‘truly more secure’.

Thus the Orange Book levels, in part, represent levels of *trust*, an unquantified belief about security. We shall show that *belief* about operational security is indeed an appropriate approach, but our intention is to find ways in which this can be formally quantified. The first task would be to try to define measures of ‘operational security’ sufficiently rigorously and realistically to gain the acceptance of the security community. This immediately presents some difficulties which, whilst possibly not insurmountable, seem more serious than is the case for reliability: we shall see that there are several open questions that may need empirical investigation.

There does not seem to be a large literature on probabilistic treatment of system security. Lee [Lee 1989] proposes to model levels of security in terms of the probability distribution of the ‘level of threat’ required to achieve a penetration involving information of a given classification. Thus $\mathrm{r}_{c_1,c_2}(t)$, denotes the probability that a system at level c_1 will allow a breach involving information classified at level c_2 when the system is subjected to a level of threat t. Hierarchical relationships between security levels can then be represented in the obvious way by inequalities between these functions. He also introduces probability distributions for the amounts of damage resulting from various illegal transfers of information. He claims that various conclusions can be drawn from this kind of model concerning the greatest security risks and the rules of combination for system security levels. However, the independence assumptions are suspect and the analysis seriously flawed in a number of other respects. Denning [Denning 1987] considers a statistical approach to detecting when the operational environment switches from ‘non-threatening’ to ‘threatening’. Bishop [Bishop 1989], in a popular account of what he calls a ‘common sense’ security model, suggests that it is possible to assign in an empirical way some of the probabilities discussed here, but gives no details.

In Section 2 we describe the terminological analogies between reliability and security, making clear the scope of our discussion. In Section 3 we discuss in detail some general problems and deficiencies associated with these analogies. In Section 4 we concentrate upon the most basic of these issues, namely the security analogy for the time-to-next-event random variable in reliability and give tentative consideration to some questions which need further investigation.

2 Terminological analogies between security and reliability

In the reliability context, the *input space* is the totality of all inputs that might ever be encountered. Thus for a process-control system it might be a many-dimensional space representing all the possible settings of sensors and controls. We can informally think of a fault as represented by a subset of this space: when an input is selected from the subset a failure will result. In the security context the input space is again the set of all possible inputs to the system, including both those involved in normal operational use *and those which result from intentional attacks upon the system.*

The *usage environment* in reliability determines the mechanism by which the inputs will be selected from the input space. The analogy of usage environment in security is thus the population of attackers and their behaviours together with normal system operation. Although we are not concerned with the detailed nature of the attacks, this population is much wider than, say, the simplistic view of a hacker typing in keyboard ‘entries’ to the system. Thus, for example, a valid attack in our sense could include

- ‘passive’ attacks, such as listening to and analysing public traffic,

- illegal use of the system, such as the insertion of trap-doors by privileged system users, or even
- attacks upon personnel, such as bribery or blackmail, that may not require any computing knowledge whatsoever, nor even any interaction with the computer system itself.

In reliability, however, it is common for the system boundary to be drawn quite tightly. Conventional hardware reliability theory tends to treat well defined physical systems; software reliability deals with failures of a well-defined program. Such a narrow view of the 'system' can be misleading [Perrow 1984] even for reliability, when, say, the people interacting with this system form a significant source of failures according to a wider view. It is common to ascribe responsibility for aircraft crashes to the pilot, for example, when more insight would be gained by viewing the pilot as a component in a wider system. The inappropriateness of this narrow view of system boundaries seems even more marked in the case of security, where often it is *precisely* the fact that attackers are interacting with the computer system and its human owners that defines the problem. We are not qualified to discuss in detail how security system boundaries should be defined in practice; suffice it to say that we intend our modelling to apply as widely as possible to practical security problems. This means that, in addition to obvious attacks on the computer system (e.g. entering password guesses), we must consider attacks upon the wider system (e.g. bribery, blackmail of personnel), and attacks which tamper with the system design before the operational system even exists (introduction of intentional vulnerabilities, e.g. Trojan horses, that can be exploited at a later stage).

The concept of system failure is central to the quantitative approach to reliability. The obvious analogy of system failure is *security breach.* By security breach we mean an event where the behaviour of the system deviates from the security requirements. We have chosen to use the word 'requirements' deliberately instead of 'policy': our intention is to capture something analogous to 'informal engineering requirements' in reliability, rather than 'formal specification', since the latter could itself be wrong. These security requirements may be stated or merely 'reasonably expected': certain events that are surprising to the specifier of the system can be considered to be failures and thus classed as breaches after the event. Our notion of security breach is therefore, like that of failure, quite general [Laprie 1989].

System failures are caused by faults in the system. Most importantly, these can be specification and design faults that have arisen as a result of human failures; indeed, in the case of computer software, this is the only source of failure. Analogously, security breaches are caused by *vulnerabilities*, such as the omission of a particular data integrity function, in the system and these are clearly special types of faults. These faults can be *accidental*, or they can be *intentional*, these latter may be malicious (Trojan horses, trap-doors) or non-malicious (resulting, for example, from deliberate trade-offs between security and efficiency). All appear to have reliability analogies except malicious intentional faults.

The notion of intentionality seems to have two meanings here. It can refer to the process whereby a system comes to have vulnerabilities (faults) - that these are inserted deliberately and sometimes maliciously during the system design. Alternatively, it can refer to the way in which those faults that are present are activated during the operational life of the system and so result in security breaches (failures). Thus we can speak of both accidental faults and intentional faults, as well as accidental failures and intentional failures. It seems that all four possible classes could exist. There is an informal, and not rigid, division into the concerns of *reliability* and *security* according to whether there is any intentionality present in the class. Reliability theory is mainly the study of *accidental* faults that result in *accidental* failures, but it must be admitted that the dividing line is not always clear and security practitioners might also concern themselves with some of these. Thus any intentional attempt to cause a breach by exploiting a vulnerability, whether this be an intentional or accidental vulnerability, is a security issue. Similarly, breaches that result from *malicious* intentional vulnerabilities, whether these are triggered by deliberate action or not, are also a security issue.

When we attempt to model these different classes, we find different problems. It seems clear that accidental security breaches, arising from accidental or intentional vulnerabilities, are the easiest to handle since they can be modelled using the techniques developed for reliability, so in this paper we shall instead concentrate on the special problems of those breaches where some intentionality is present in the *failure* process. Perhaps the hardest class is that of intentional breaches arising from intentional malicious vulnerabilities, of which Trojan Horses are an example. It may be possible to model the process of *insertion* of such vulnerabilities, but modelling the process by which they are triggered during system operation looks difficult.

The most difficult analogy is with that of *execution time* in reliability. In the reliability context we are always dealing with a stochastic process, usually of *failures* in *time.* Although in different contexts we would wish to express the reliability differently (e.g. ROCOF, mean time to failure, probability of failure-free mission), a complete description of the underlying stochastic process would be sufficient for all questions that are of interest. Even in the reliability context, care needs to be taken in choosing the time variable to be used. In the case of a nuclear reactor protection system, for example, it might be convenient to express the reliability as a probability of failure upon demand rather than as a rate of occurrence of failures in (calendar) time. Although the latter is of interest as a measure of the reliability of the wider system which incorporates this safety system, it would be unreasonable to allow the frequency of demands to affect our judgement of the reliability of the protection system. Here time is a discrete variable, represented by the individual demands. The important point is that there is always some notion of stress, or load, involved in our choice of time variable, even though this may be done quite informally. Occasionally it may be the case that calendar time itself will be a good representation of this stress, for example in certain types of process control. Usually, however, we need at least to be careful to count only the time that the system is being exercised, so that in the case of software reliability *execution time* has come to be a widely used variable [Musa 1975].

For the security ‘process’, where we are trying to model the process that results from intentional attacks, it is not obvious what should play the role of time in order to capture the notion of stress. Clearly a system might be able to continue to deliver its ordinary functionality indefinitely without security breaches, as long as there is no attack present. This contrasts with the case in reliability, where it is usually assumed that a system is always susceptible to failure as a result of accidental faults (even though the extent of the susceptibility, expressed for example as a failure rate, will depend on the nature of the use). It seems that time, however ingeniously defined, will rarely be an appropriate variable in the case of intentional breaches. Instead we think it is necessary to consider the *effort* expended by the attacking agent. This effort *could* sometimes be time itself, perhaps the time expended by the attacking agent, but it will be only rarely the same time as that seen by, say, the system user or owner. More usually, effort will be an indirect measure of a whole range of attributes, including financial cost, elapsed time, experience and ability of attacker, etc. In particular, such an effort measure can take account of such behaviour as learning about a system off-line, or by using an entirely different system from the one that is the subject of the attack. It is interesting that effort has previously been used informally in security. For example it is often suggested that the effectiveness of a cryptographic technique is measured in terms of the cost of accessing the data in comparison with the value of that data to the attacker.

Using effort may also allow us to deal with two situations where the reliability analogy is not helpful. In the first place, there is the case of intentional breaches arising from intentional malicious faults, such as Trojan Horses. Here any notion of time of usage of the system is meaningless: the effort is expended in inserting the intentional fault, before the operational system exists, and any effort involved in triggering such a latent fault to cause the security breach during operational use is likely to be trivial. Secondly, there are circumstances where the expenditure of effort is instantaneous, and any notion of sequence, which seems essential in a time variable, is absent. An example would be the offering of a bribe to exploit human cupidity; here the chance of success would clearly depend upon the magnitude of the bribe (effort), but it is hard to see this as analogous to time.

Finally, if we accept that effort in security can play the same role that a general notion of time plays in reliability, then it is reasonable to consider the *effort-to-breach distribution* as analogous to the *time-to-failure distribution* of reliability. Assuming that we can obtain an appropriate variable to represent effort, operational security could in principle now be expressed in terms of probability statements: for example, mean effort to next security breach, probability of successfully resisting an attack[1] involving a given expenditure of effort, etc.

In practice, of course, from most viewpoints it *is* often desirable to know something about the security of a system expressed in terms of time: a system owner, for example, might like to know the probability that there will be no breaches of the system during its predicted lifetime. Clearly this requires knowledge of a doubly stochastic process: in the first place we need to understand the process of security breaches as a function of effort, then we need to know the process of effort as a function of time. In this paper we are going to restrict ourselves to the former: in statistical terminology we shall be making probability statements that are *conditional* upon the effort process. Although one justification for this approach is pragmatic, specifically that knowledge of the effort process (in time) is hard to obtain and very much dependent upon the application and the nature of the environment, there is a more important reason. Seeing security *conditionally* as a system attribute (informally its ability to resist attacks involving a specified effort environment) seems more natural than seeing it *marginally,* that is averaging over all the different efforts (environments) that attackers *might* expend.

We have addressed this issue at some length since it seems to be an important departure from the reliability analogy. The departure is not, of course, an absolute one since the two stage approach could also be applied in reliability: the failure process as a function of system load, and the load process as a function of time. Indeed, in some reliability work exactly this structure is adopted. More usually, however, the load process is simply equated with a suitably defined time variable, and this can often be a good

[1] Notice that we now have a broad definition of attack that includes any expenditure of effort with malicious intent.

approximation to reality. In security, we believe, such an approximation via time alone is rarely valid.

These detailed remarks apply to intentional breaches [Laprie 1989]. As we have seen, accidental breaches will result in a process of security breaches similar in its properties to the failures treated by reliability methods involving *time.* It is clear from the above that we could not model in a single process *all* security breaches, accidental and intentional, without having knowledge of the effort process in time.

It is worth remarking here that security work seems to have concentrated to a large extent on the problems arising in circumstances where *very* high security is demanded. Certainly it appears to be the case that some of the major funding agencies for security research are concerned with issues of national security. Problems of this kind in security are similar to the problems of safety-critical systems in reliability, where very high levels of reliability often need to be assured. The work on ultra-high reliability evaluation, however, has so far been notably unsuccessful, at least for software and design reliability [Littlewood 1991], in comparison with the achievements in the modelling and evaluation of more modest levels [Brocklehurst and Littlewood 1992]. This lesson suggests that, in these initial stages of attempting to model operational security, we should restrict ourselves to systems for which the security requirements are also relatively modest. It is only for such systems that sufficient numbers of breaches could be observed for the empirical studies that seem necessary.

3 Difficulties and deficiencies of the analogies

3.1 Failures, security breaches and reward processes

In reliability we observe a realisation of a *stochastic point process* of failures in time, Fig 1. It is sometimes convenient to show this graphically as the step function representing the cumulative number of failures against time, Fig 2. Such a plot will usually clearly reveal the growth in reliability, due to fault-removal, as the failures tend to become more separated in time and the ‘slope’ of the plot becomes smaller. In principle, in the case of security, a similar plot could be drawn of the number of security breaches against effort.

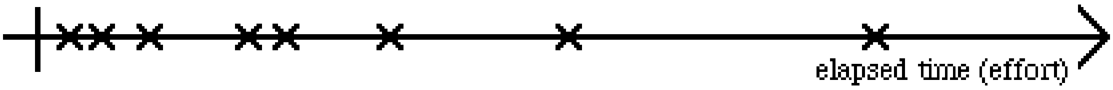

Fig. 1 The failure process in reliability (breach process in security).

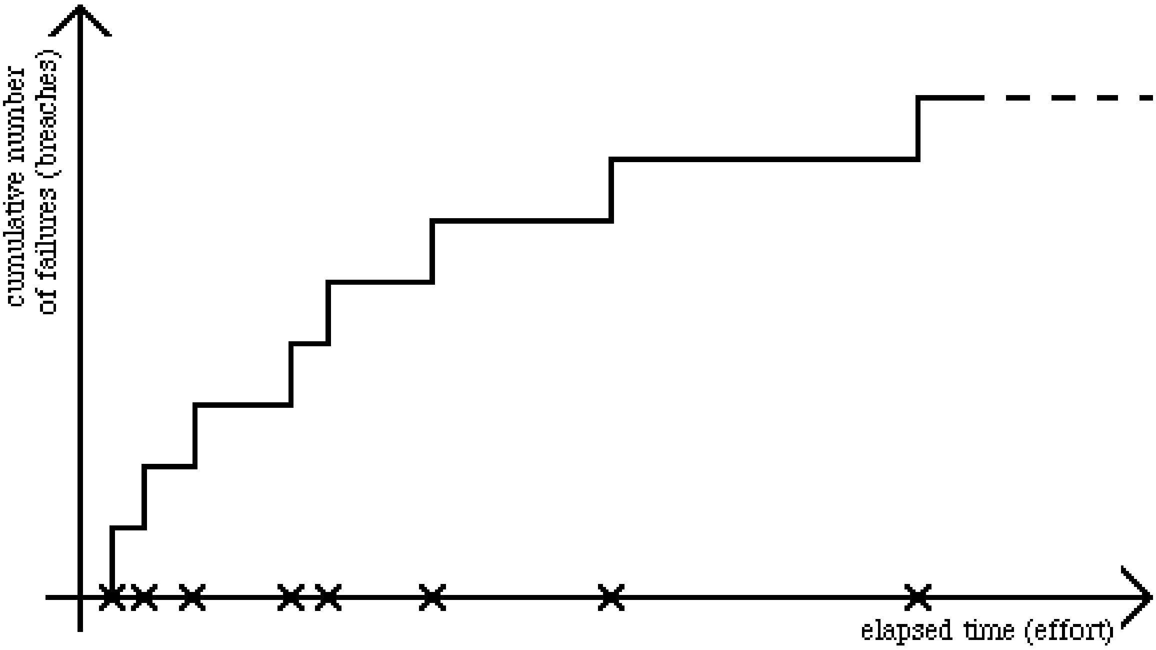

Fig. 2 Typical cumulative failure plot for reliability data (security equivalent).

Even in reliability, such a view of the failure process can be too simplistic. A user is interested not only in the process of failures, but also in their (random variable) consequences. In security especially it is likely to be the case that not all security breaches have the same value for the attacker (similar remarks apply to the consequences for the system owner). The reward an attacker would get from breaking into a system determines her[2] motivation and affects whether she is willing to spend the effort that is needed in order to perform the attack. Examples of rewards are personal satisfaction, gain of money, revenge or simply pure curiosity, but it is also possible that the attacker may get negative rewards (losses) such as those due to the consequences of detection. A more general model, then, would describe the stochastic process of *rewards* to the attacker as she incurs effort, Fig 3.

[2] For convenience we adopt the convention that the system owner is male, the attacker female.

There is a further generalisation in the security context that does not seem to have a parallel in reliability. That is the possibility that an attacker may gain continuous reward even in the absence of security breaches. Consider, for example, the very simple case of a system with a finite number, n, of potential passwords, and an attacker trying these at random but not repeating failed attempts. Clearly, when she has tried m passwords, all of which have failed, she has learned something that is of value. In a sense, this value is only potential, since she has not actually acquired whatever benefit will accrue from the actual breach (for example, changing financial software in order to steal money). But this is value nevertheless: thus, for example, she could sell this information to another attacker, who might then thereby expect to be able to gain entry with less expenditure of effort. Indeed, it seems clear that the (potential) value of these failures approaches the value of a breach as $m \to n$. In cases such as this we need to consider the possibility that reward may increase essentially continuously as effort is expended, as well as discretely at particular events (e.g. security breaches), Fig 4.

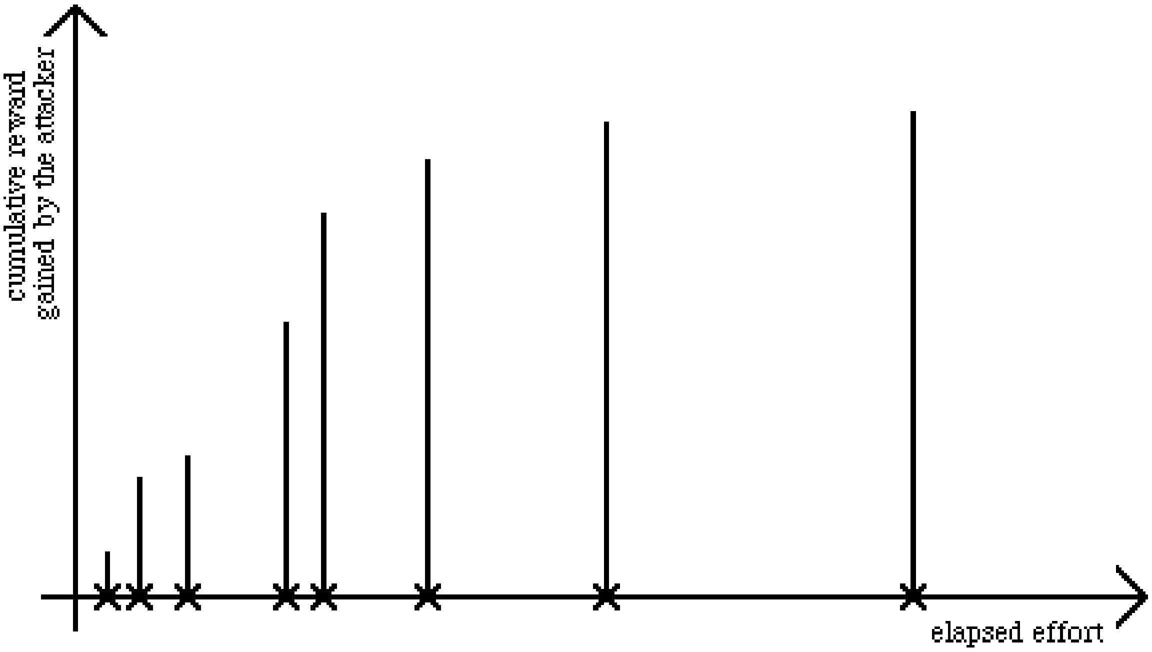

Fig. 3 Stochastic process of rewards to the attacker as she incurs effort; reward obtained in discrete amounts only as the result of a breach.

In summary, there are three levels at which operational security modelling could be aimed: a simple counting process of breaches versus effort, a reward process of random variables defined on these events, and the most general random process of reward as a function of effort expended as well as events observed. These are clearly in ascending order of realism, as well as of difficulty (particularly with respect to the availabililty of empirical data). For example, it is clearly easier to recognise the mere fact of a security event than to estimate the reward associated with such an event, not least because of its subjective nature.

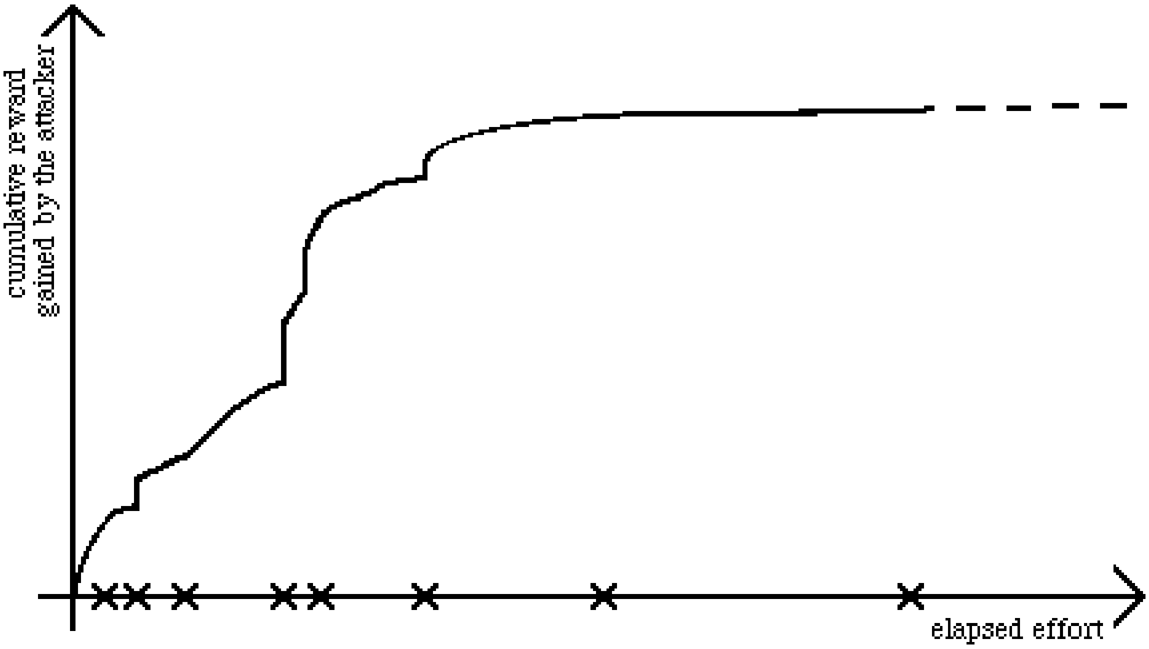

Fig. 4 Stochastic process of rewards to the attacker as she incurs effort; here reward may be obtained continuously as effort is expended.

3.2 Usage environment analogy

It is well known that the reliability of systems depends upon the nature of their use. Strictly, therefore, there is no sense in which we can really speak of 'the' reliability of a system, but instead we ought to specify the exact nature of the operating environment and a reliability figure will then refer to both the system and this environment. In other words, we should always be careful in our probability statements to specify the exact nature of the conditioning events. Thus reliability could be expressed as a mean time to failure *given that* the system is operating in environment A, security could be expressed as the mean effort to security breach *given that* the system is operating in threatening environment B.

Here we are addressing a slightly different point from our earlier discussion of effort versus time. There we argued that the effort process in time was an attribute of the environment, and we showed that there were advantages in treating this separately from the process of security breaches as a function of effort. However, effort cannot capture all the variation that there will be between different environments. One example, of course, is that variation in the nature of the *legitimate* use of the system may increase its susceptibility to security breaches, for a particular fixed effort expenditure on the part of an attacker. This is similar to the reliability case, but in addition there is variation arising solely from the presence of attackers.

Unfortunately, it seems possible that there is more variability in threatening environments in security than there is in operational environments in reliability, not least because the threatening environments may be deliberately created rather than randomly generated. We are dealing with deliberately malign attackers rather than merely randomly perverse nature[3]. In the software reliability context, for example, we assume that in principle it is possible to identify *a priori* all possible inputs, even though this may be impracticably onerous in a particular case. In the case of security, it is not clear that even in principle it is possible to identify all possible attacks *a priori*. Sometimes, in fact, the role of the attacker is to *invent* that which has not been thought of. Will such an attacker restrict herself to trying out attacks from an initial repertoire, modelled analogously to some fixed input space? Or is it more realistic to assume some kind of learning process and increased familiarity with the system, until entirely new attacks will be invented which exploit peculiarities of the particular system, and were initially completely outside the imagination of the attacker (and system owner). If this is the case, then it represents a kind of enlargement of the space of potential attacks over time, or at least a drastic shift in the operational profile (probability distribution over the input space) resulting in attacks which were initially unimagined (i.e. ‘impossible’) acquiring an increasing probability of occurrence as time and effort are expended.

Clearly, one effect of intentionality and learning would be the tendency eventually to give up on a line of attack which had proved unsuccessful once its possibilities appeared exhausted. The probability attached to the corresponding subset of the attack space would in this case decrease over time. This means that we shall never encounter in security the situation that pertains in reliability of a fixed operational profile.

Such considerations seem to have at least two sets of implications for any model of effort-to-breach random variables. In the first place, the properties of the process (against effort) of breaches caused by an individual attacker will be affected, for example the hazard rate of the effort-to-next event distribution. Secondly, the rules of combination for the breaches achieved by different attackers will be affected by considerations such as whether they communicate and learn from each other’s

[3] Notwithstanding the differing likely variability in environments, it is not clear whether human ingenuity or nature can produce the *greatest* perversity (variability). Inasmuch as we all, as humans, share much of our cultural baggage, one view would be that nature, at its most extreme, will occasionally produce those results that surprise us the most.

experience, and whether, after repertoires of standard attacks have been exhausted, their imaginations tend to diverge in thinking of novel forms of attack.

Having said all this, it is not clear that these considerations are overriding. For example, it does seem a rather subtle distinction that is being made above between the security scenario, where the input space of attacks is expanded by the inventiveness of the attacker, and the reliability scenario, where the 'perverseness of nature' throws up inputs that we *could have* anticipated but *didn't*. In both cases it seems plausible to assume that the view of the system owner, for example, will change in the same way: he merely sees inputs being generated that he had not thought of. This is reminiscent of the discovery of specification faults during operational use. Perhaps such considerations do not detract from the principle of treating security stochastically, but only suggest we should be wary of the details. Once again, empirical investigation may be the best way forward to decide upon some of these open questions.

3.3 Viewpoints

In all of the above it is taken for granted that *probability* is a suitable way of expressing our uncertainty about how a system responds to its operating environment, in both the reliability and security contexts. In the case of reliability this is probably more obvious. There we can think of the system receiving a sequence of input cases from some (usually large) input space. Some of the inputs in the space are such that, when they are presented to the system, unacceptable output is produced: we say the system has failed. There is uncertainty about which inputs will be selected in the future, and there is uncertainty about which untried inputs will produce unacceptable output. The first source of uncertainty relates to the aforementioned nature of the operational environment, and it is to be expected that if we inspected streams of inputs from two different environments we would be able to observe differences. It has been argued elsewhere [Littlewood 1979; Littlewood 1991] that this uncertainty should be represented by probability, and that Bayesian subjective probability seems most appropriate.

It should be emphasised that the subjective interpretation of probability is not in any way less 'scientific' than the more usual interpretation based on limiting relative frequency. The present paper is not the place to argue this case; interested readers should consult [Finetti 1975; Lindley 1985].

This approach via the subjective interpretation of probability seems particularly appropriate in the case of security. Here we must contend with the possibility of more than one viewpoint of the system. The system owner, even if he could be regarded as having complete knowledge of the system (i.e. of those inputs that would result in breaches), is uncertain about the sequence of inputs (here attacks) that will arise from the adversary's attempts to breach security. This adversary, on the other hand, is at least partially ignorant about the system and thus does not know the potential effects of her possible choices of inputs (attacks), even though she has complete knowledge and control of these choices. It seems clear that, since they are based upon information which is incomplete in different ways, these two subjective views need not coincide; indeed it would be surprising if they did. In fact, not only will there be different probabilities associated with the same event when seen from the different viewpoints, but some events may not be seen from all viewpoints - for example, some breaches may not be seen by the system owner. In addition to this, *rewards* are also likely to differ from one viewpoint to another, even when the same events are observed. The attacker's reward resulting from a breach will not generally be the same as the system owner's loss[4]: there is no sense in which this is a zero-sum game.

It may be helpful to consider other viewpoints. Thus, for example, in security testing, it may be sensible to consider the viewpoint of an all-knowing, all-seeing oracle, 'God', as well as the owner and attacker. God would see all security breaches, including those not detected during test by the owner (and those not detected by the attacker), and so this viewpoint could be regarded as being in a sense the 'true' security of the system in the testing environment. If the owner could trust the security test to be representative of the operational (threatening) environment, it would be interesting to him to compare his own subjective view of the system's security with God's. Thus if the system owner knew that in *test* he had formed an unreasonably optimistic view of the security of the system, it would be prudent of him to regard his view of the *operational* security as also optimistic. It might even be possible for him to carry out a formal recalibration of his subjective view of the operational security, using this information from testing together with the operational observations.

3.4 System boundaries and definition

In system reliability we normally have a very clear idea of the boundaries of the system to which we are to attach reliability measures. Usually this is a well-defined system, operating in a well-understood environment, and the important variable upon which the failure process is defined is *time.*

[4] Information seems to be unique in that you (usually) still own that which you have given away, sold, or had stolen; but its value to you has changed as a result of such a transaction.

For security evaluation we should ideally view the operational system, which is the overt subject of the security evaluation, in the widest sense. Thus we should include all products associated with the system: that is all documentation (including, for example, requirements specification, designs and test results) and all executable code. We should include all resources associated with the development and operation of the system: for example, personnel, software tools such as compilers, and hardware.

As we have seen earlier, for security it is convenient to define the relevant process instead upon *effort*. Although in many cases it is still valid to use the reliability analogy of inputs to this system being prone to result in security breaches, the use of effort instead of time allows us to generalise the notion of attack. Although we have so far been largely concerned with the modelling of intentional *breaches*, it may also be possible to use effort to model the introduction of intentional *vulnerabilities* during system development, such as Trojan horses. In a case like this, it becomes difficult to argue that it is the operational system that is under attack. On the other hand, the consequences of such an attack can certainly show themselves during operational use of this final product.

One way forward might be to consider this kind of security measure as different in kind from the measures for the operational system that have been the subject of the earlier discussion. Thus we might, for example, try to estimate the probability that the system is free from malicious intentional vulnerabilities; this could be seen as a measure of the 'security' of the development process. Although ideally we would like to know the effect of potential malicious intentional vulnerabilities upon operational security, so as to arrive at a single measure incorporating all possible breaches, this seems a sufficiently hard problem that the foregoing separation of concerns might be the best approach at this stage.

4 Probabilistic requirements for an operational security model

In the remainder of the paper, for simplicity, we shall not consider these malicious intentional vulnerabilities. Neither shall we deal with accidental breaches since, as we have discussed earlier, these seem most easily amenable to the reliability approach. Instead, in this section we shall restrict ourselves to some important questions concerning a probabilistic treatment of intentional attacks upon a system containing accidental (and non-malicious intentional) vulnerabilities. A further simplification here is that we shall not treat reward processes; instead we shall concentrate upon breach event processes since an understanding of these is a necessary precursor to any modelling of rewards.

There are three basic scenarios that we might consider. The first of these concerns a system that has been exposed to security attacks, but has not suffered a breach. The second concerns the case where breaches have occurred and there have been attempts to correct the underlying vulnerabilities. These situations correspond, respectively, to the work on estimating the reliability of a program that has never failed [Littlewood 1991], and to the classic reliability growth models [Abdel-Ghaly, Chan et al. 1986; Jelinski and Moranda 1972; Littlewood and Verrall 1973; Musa 1975]. The intermediate case in reliability, where failures occur but faults are not fixed, can also arise in security - for example for economic reasons in a system of only modest security. Unfortunately, whereas in reliability this produces a particularly simple, stationary stochastic process, in security things are more complicated. Some vulnerabilities are of a nature that will allow many breaches to follow the first one with little effort, and a stochastic process of clusters of breaches will result.

These three cases all concern situations where any attempt to remove vulnerabilities arises solely from the discovery of these via detection of security breaches. Clearly it is also possible for vulnerabilities to be discovered before they have been successfully exploited by real attackers. One important example of this is, of course, security testing (although it *may* be possible to use data from such testing to obtain estimates of operational security, and this will be discussed below). Other methods of discovering vulnerabilities, such as design reviews, code inspections, etc, do not readily lend themselves to modelling operational security, as is equally the case for their equivalents in reliability. Similar problems arise in modelling the effects of vulnerabilities which are removed in operational use without their having been revealed by breaches, for example by the system manager noticing unusual patterns of use. It may sometimes be the case that countermeasures are known for certain types of attack but these have not been implemented for economic reasons; if a particular system is found to be operating in an environment which exposes it to this kind of attack, it may be possible to quickly insert protection and avert a breach. We shall not attempt at this stage to address any of these issues. It is worth noting, however, that neither do the classical reliability models address the equivalent reliability problems.

Before considering the complication of treating the whole stochastic process of successive breaches (and consequent evolution in the security), we need to understand the simpler case concerning a snapshot of the security at a single moment in time. This concerns the distribution of the (random variable) effort, E, required for the next breach[5]. We begin with some notation. The *security function*, by analogy with the *reliability function,* is

$$\mathrm{R}(e) = \mathrm{P}(E > e)$$

with the cumulative distribution function (cdf) and probability density function (pdf), if it exists, of effort to next breach

$$\mathrm{F}(e) = 1 - \mathrm{R}(e), \qquad \mathrm{f}(e) = \mathrm{F}'(e)$$

respectively, and *security hazard rate* (cf *hazard rate*)

$$\mathrm{h}(e) = \mathrm{f}(e)/\mathrm{R}(e)$$

We shall generally wish to make clear the particular viewpoint, in which case we shall denote attacker's security function $\mathrm{R_a}(e) = \mathrm{P}(\mathrm{E_a}<\mathrm{e})$, owner's $\mathrm{R_o}(e) = \mathrm{P}(\mathrm{E_o}<\mathrm{e})$, etc.

The equivalent problem in reliability concerns the distribution of the time to next failure of a program. The simplest assumption here is that this distribution is as it would be if the failure process were 'purely random'. Then there is a constant probability $\lambda.\delta t$ of a failure in a short time interval $(t, t + \delta t]$, independently of what has happened in $(0, t]$, i.e. the process is memoryless. Here λ is the failure rate of the exponential distribution of T, the random variable time to next failure. This assumption seems reasonably plausible in the software reliability context, for example, where T represents the time spent in a trajectory through the input space before meeting the next input to precipitate a failure. In practice, of course, the rate λ is never known exactly, and so the exponential distribution of T should be seen only as a conditional one; to express our unconditional beliefs about T we need in addition to describe our belief about λ.

In reliability, the exponentiality can be seen as the limiting case of the number of input cases until the next failure. In the simplest case, there is a constant probability p of failure on each input, independently for successive inputs. The number of inputs until next failure is then geometrically distributed, which is exponential in the limit as $p \rightarrow 0$. This limit applies under much wider conditions: the probabilities of failure on the different inputs do not need to be equal, the successive inputs do not need to be independent with respect to failures. We can expect the time to next failure to be approximately exponential as long as there is *stationarity,* i.e. no tendency for the

[5] This breach must be the first one associated with a particular vulnerability.

probability of failure on an input to be a function of time, and there is limited dependence between successive inputs: both these conditions seem plausible for real systems.

Can any of these ideas be carried over into the security context? In particular, are there circumstances where it is reasonable to expect the ‘effort to next security breach’ random variable to be conditionally exponentially distributed? The question arises as to whether similar underlying conditions hold in the security context to make the exponential assumption plausible there. Certainly, it is not clear whether the necessary assumption of stationarity holds in the security context.

Consider the viewpoint of the attacker. If we regard each attempt to carry out a security breach as a series of elementary ‘attacks’, it seems likely that these attacks will be carried out in such a way as to minimise the perceived likely effort of success. Thus, for example, the attacks may be carried out in a descending order of (perceived) chance of success (per unit outlay of effort). From the attacker viewpoint, then, there is a clear violation of the stationarity condition needed for exponentiality of the attacker’s subjective probabilistic distribution of the random variable ‘effort’, E_a, until next breach. We might expect the distribution to be a ‘decreasing failure rate’ (DFR) one [Barlow and Proschan 1975]. The system owner’s view of such a process of attacks from a single attacker is also likely to be DFR, but less strongly so if he sees a sequence of attacks that differs in its ordering from the optimal ordering that he (the owner) believes to be the case.

The system owner may sometimes see the combined effects of many attackers operating simultaneously, and their *different* rankings of likely successful attacks will again tend to diminish the tendency for DFR-ness. However, the system owner will presumably believe there is some positive correlation between different attackers’ views of likely successful attacks, suggesting that this viewpoint *will* also be DFR.

Any tendency for DFR from the owner’s viewpoint may also be weakened in the following way. If an owner is to see many attackers, it is most likely that these will not make their attacks simultaneously, but approximately sequentially. Thus at any one time it may be the case that there are a few, but not many, attacks under way. Since it might also be reasonable to assume that most attackers incur the expenditure of a relatively small amount of effort, as a crude approximation we could regard the process as a stationary sequence of many different successive attacks as a function of the total effort incurred by the population of attackers, E_o. In a case like this, then, the system owner might be justified in believing the distribution of E_o to be exponential.

The arguments here do not, however, take account of any learning about a system's weaknesses that an attacker could gain from her successive failures to breach the security, or from the failures of other attackers. Presumably such learning will tend to *increase* the success rate, producing, in the absence of any contrary tendency such as mentioned above, an increasing failure rate (IFR) distribution for the effort to next breach. These remarks apply to the attacker's viewpoint, and are probably applicable in a weaker form for the owner viewpoint, using similar reasoning to above.

It may be the case that *both* these tendencies - to DFR-ness and to IFR-ness - are present, and tend to cancel one another out. In that case the exponential may be a reasonable approximation for distribution of the time to next breach. At this stage, the important point is that *we do not know* the answers to questions such as these, and they are thus candidates for empirical investigation.

Indeed, concentration upon exponentiality versus IFR or DFR could involve over-simplification, since the actual distributions here might fit into none of these classes. We now consider some more general questions about these distributions that are prompted by the special circumstances of the security context.

One interesting question concerns what happens as the amount of effort expended becomes large. Clearly R(e) decreases as e increases, but can we assume it approaches 0, i.e. that breach is certain with the expenditure of unlimited resources? If instead it approaches a non-zero limit as e becomes infinite, the distribution of effort to breach is 'improper' - there is a chance that even with infinite expenditure there will be no breach. This seems to be a plausible description of what an attacker would believe and would therefore be an argument against exponentiality.

Another issue concerns the several ways in which 'combination' plays an important role in security. For example, for a single attacker, the E_a random variable is presumably really a sum of the efforts of 'elementary' failed attacks: the implementation of different ideas as early ones are tried and fail, stopping with the first success. What does a sequence of elementary attacks look like? We have already touched on the question of whether the process of attack outcomes against effort is stationary, or monotonically increasing or decreasing. Could there be more complex behaviour here? What, for example, is the nature of the dependence between the outcomes of successive attacks? Independence seems too simplistic, if only because an attack that has failed will not be tried again, but presumably the dependence drops off with distance. Is there a law of large numbers here: *large* number of *small* attacks, each with a *low* probability of success? How should the stopping rule for an attacker operate? Is it too naive to assume she has infinite resources? Or should we assume finite resources? If so, does this affect her allocation and ordering of the elementary attacks? What different kinds of objective function might an attacker employ in identifying one particular ordering as optimal?

Similar issues of combination arise when we consider how the owner would see a *combination of attackers*? This involves us in a generalisation of the univariate distributions considered earlier. Assume that the owner knows that there are *n* attackers. Let $R_o(e_1, . . . , e_n)$ represent the probability that the system has not been breached when, for i = 1, . . n, the *i*th attacker has expended e_i. What does $R_o(e_1, . . . e_n)$ look like?

A very naïve approach assumes that attackers are independent, and similar, with successes occurring purely randomly, i.e. $R_o(e_1, . . . e_n)$ = exp(-lSe_i). The likelihood function, when we have seen no breach in many attacks, is then exp(-lSe_i), i.e. the same as a single attack of expenditure Σe_i. This is not very interesting! It implies that a large number of small attacks is equivalent to a small number of large attacks. Is this reasonable? If not, can we frame the optimality issues formally: if there was a certain amount of effort, Se_i, to be expended, what allocation of this among attackers would the owner fear most? And how would this change with Se_i*?*

Are there *any* circumstances where the owner might be interested only in the total effort being expended, and not at all on its allocation (this is, in the statistical terminology, a kind of ‘sufficient statistic’ argument)? Do the hardware shock models [Barlow and Proschan 1975] have any relevance here: is there a sense in which the system succumbs as a result of the *accumulation* of attacks (for example via accumulation of knowledge), or is it waiting for an attack beyond some suitably defined *threshold*?

If, as seems to be the case, we cannot assume independence between attackers, what is the nature of the dependence? One view might be that the different attackers are selecting their ‘elementary attacks’ from the same fairly small finite set: is this reasonable?

Is it reasonable, at least, to assume that $R_o(e_1, . . . e_n)$ is ‘exchangeable’, i.e. that it remains unchanged under permutation of the subscripts, representing a kind of subjective indifference between different attackers on the part of the system owner?

If we consider the way that a group of cooperating attackers might divide up the effort amongst themselves, there arises the intriguing possibility of modelling something like the ‘forced diversity’ ideas in Littlewood and Miller [Littlewood and Miller 1989]. Here the Eckhardt and Lee [Eckhardt and Lee 1985] work on program diversity through separate and ‘independent’ development was generalised to introduce the idea of ‘negatively correlated’ development. It can be shown that in principle such forced diversity can give better results in a fault tolerant system than those achievable under (anyway unattainable) independence. It was also shown that there was a complete duality in this conceptual modelling: for every theorem about many program versions operating in a single operational environment, there is a dual concerning a single program operating in many operational environments. Is it reasonable to think that there are similar results to these in the security context? For example, does it make sense to talk of forced diversity in co-operating attackers?

5 Conclusions

In this paper, we have begun to address some quantitative aspects of operational security in a manner analogous to that which has been successful for operational reliability. We have identified some important questions which need to be answered before this quantitative approach can be taken further. We have mainly restricted ourselves to the very simplest problem, analogous to the situation in reliability concerning time to next failure. There are clearly other parallels worth investigating. For example, we have not yet considered any of the interesting parallels with *availability*. We hope to address some of these issues in further work, but we believe that the difficulties posed even by the restricted scenario presented here need to be resolved before we can attempt to model these more difficult cases.

At the very least, we hope that the preceding discussion has made clear the desirability of a probability-based framework for operational security measurement. However, the many unanswered questions that have arisen in the course of the discussion (and we would not claim that those raised here are exhaustive) mean that we are presently far from having a modelling theory that would allow us to allocate quantitative measures of operational security to particular systems, and it is this which will determine whether this new approach is of practical utility. Where do we go from here?

Some of the open questions are suitable for empirical investigation, and one way forward might be to devise controlled experiments in which ‘tame’ attackers are allowed to try to make security breaches on a designated system. A combination of system instrumentation and attacker interview might allow us to learn about the key notions of effort and reward that have been discussed here. Some of the authors of this paper are presently conducting a pilot experiment along these lines. Another approach would be to collect data on real systems in operation, as is done in the case of reliability studies. This seems fraught with difficulty, but it is a possibility that we do not rule out.

A different approach, and one of our reasons for writing this paper, would be to ask whether there is any consensus within the security community on any of these questions. Since our approach acknowledges the basically subjective nature of many of the questions - it concerns the way in which an individual will construct a quantitative personal viewpoint - it would be valuable to learn what practitioners currently think about these issues. We would therefore welcome comments from members of the security community on the feasibility of this whole approach, as well as answers to the specific questions we have raised.

Acknowledgements

This work was carried out as part of the ESPRIT projects 3092 (PDCS) and 6362 (PDCS2). Earlier versions of the paper were read by Tom Anderson, Chris Dale, Yves Deswarte, Jean-Claude Laprie, David Powell, Michael Roe, Lorenzo Strigini. We are grateful for their comments and suggestions, and for those provided by the responsible editor and the reviewers.

References

[Abdel-Ghaly, Chan et al. 1986] A.A. Abdel-Ghaly, P.Y. Chan and B. Littlewood, "Evaluation of Competing Software Reliability Predictions," *IEEE Trans. on Software Engineering,* vol. SE-12, no. 9, pp.950-967, 1986.

[Barlow and Proschan 1975] R.E. Barlow and F. Proschan. *Statistical Theory of Reliability and Life Testing,* New York, Holt, Rinehart and Winston, 1975, 290 p.

[Bishop 1989] R. Bishop, "Computer security - a common sense model," *Computer,* no. 5 October, pp.42-43, 1989.

[Brocklehurst and Littlewood 1992] S. Brocklehurst and B. Littlewood, "New ways to get accurate reliability measures," *IEEE Software,* vol. 9, no. 4, pp.34-42, 1992.

[Denning 1987] D.E. Denning, "An intrusion-detection model," *IEEE Trans Software Engineering,* vol. SE-12, no. 2, pp.222-232, 1987.

[Eckhardt and Lee 1985] D.E. Eckhardt and L.D. Lee, "A Theoretical Basis of Multiversion Software Subject to Coincident Errors," *IEEE Trans. on Software Engineering,* vol. SE-11, pp.1511-1517, 1985.

[Finetti 1975] B. de Finetti. *Theory of Probability,* Chichester, Wiley, 1975.

[Jelinski and Moranda 1972] Z. Jelinski and P.B. Moranda. "Software Reliability Research," in *Statistical Computer Performance Evaluation,* pp. 465-484, New York, Academic Press, 1972.

[Laprie 1989] J.-C. Laprie. "Dependability: a unifying concept for Reliable Computing and Fault Tolerance," in *Resilient Computing Systems,* pp. 1-28, Oxford, Blackwell Scientific Publications, 1989.

[Lee 1989] T.M.P. Lee. "Statistical models of trust: TCBs versus people," in *IEEE Symposium on Security and Privacy,* pp. 10-19, Oakland, IEEE Computer Society Press, 1989.

[Lindley 1985] D.V. Lindley. *Making Decisions,* Chichester, UK, Wiley, 1985.

[Littlewood 1979] B. Littlewood, “How to Measure Software Reliability, and How Not to ...,” *IEEE Trans. on Reliability*, vol. R-28, no. 2, pp.103-110, 1979.

[Littlewood 1989] B. Littlewood, “Predicting software reliability,” *Phil Trans Royal Soc London*, vol. A 327, pp.513-49, 1989.

[Littlewood 1991] B. Littlewood. “Limits to evaluation of software dependability,” in *Software Reliability and Metrics (Proceedings of 7th Annual CSR Conference, Garmisch-Partenkirchen),* pp. 81-110, London, Elsevier, 1991.

[Littlewood and Miller 1989] B. Littlewood and D.R. Miller, “Conceptual modelling of coincident failures in multi-version software,” *IEEE Trans on Software Engineering,* vol. SE-15, no. 12, pp.1596-1614, 1989.

[Littlewood and Verrall 1973] B. Littlewood and J.L. Verrall, “A Bayesian Reliability Growth Model for Computer Software,” *J. Roy. Statist. Soc. C*, vol. 22, pp.332-346, 1973.

[Musa 1975] J.D. Musa, “A Theory of Software Reliability and its Application,” *IEEE Trans. on Software Engineering*, vol. SE-1, pp.312-327, 1975.

[NCSC 1985] NCSC. *Department of Defense Trusted Computer System Evaluation,* DOD 5200.28.STD, National Computer Security Center, Department of Defense, 1985.

[Perrow 1984] C. Perrow. *Normal Accidents: Living with High Risk Technologies,* New York, Basic Books, 1984.