Optimistic Security: A New Access Control Paradigm

Dean Povey
Cooperative Research Centre for Enterprise Distributed Systems*
povey@dstc.edu.au

Abstract

Despite the best efforts of security researchers, sometimes the static nature of authorisation can cause unexpected risks for users working in a dynamically changing environment. Disasters, medical emergencies or time-critical events can all lead to situations where the ability to relax normal access rules can become critically important.

This paper presents an optimistic access control scheme where enforcement of rules is retrospective. The system administrator is relied on to ensure that the system is not misused, and compensating transactions are used to ensure that the system integrity can be recovered in the case of a breach. It is argued that providing an optimistic scheme alongside a traditional access control mechanism can provide a useful means for users to exceed their normal privileges on the rare occasion that the situation warrants it.

The idea of a *partially-formed* transaction is introduced to show how accesses in an optimistic system might be constrained. This model is formally described and related to the Clark-Wilson integrity model.

1 Introduction

Bob is a nurse at a small rural hospital which has been physically isolated due to a heavy storm. Communications are down, and the local doctor is unable to be located. Bob has to attend to a life-threatening emergency, for which he needs immediate access to a patient's medical records. However, Bob is not authorised to access the information, putting the patient's life at risk.

Regardless of how flexible or expressive access control mechanisms become, there will always be a gap between what organisations need, and what mechanisms can implement. One reason for this is that access control systems are unaware of dynamically changing situations that are external to the system.

Alice is working late one night, when she notices a rogue process is going crazy and forking off multiple children. Alice knows that if she doesn't kill the process quickly, it is going to eventually kill the machine, bringing down a mission-critical system in the middle of its nightly batch scripts. However, Alice doesn't have superuser privileges on the box, so all she can do is wait and watch as the system begins to fall over.

Whether it's a disaster, medical emergency, or just a need to meet a critical deadline, sometimes the necessary authorisation depends on situations which are unforeseen, and which a system cannot be easily made aware of. An organisation that wishes to define a security policy which differentiates roles depending on circumstances, will find themselves unable to implement the policy using traditional schemes which enforce least privilege.

This paper investigates an optimistic access control scheme as a new paradigm for constraining access in such situations. Optimistic access control takes the approach of assuming that most accesses will be legitimate, and relies on controls external to the system to ensure that the organisations security policy is maintained. The scheme allows users to exceed their normal privileges in a way which is constrained so that it is securely audited and may be rolled back. Ways of minimising risks in an optimistic access control scheme are discussed, and a formal model is given based on the Clark-Wilson Integrity Model[2]. Finally, the paper concludes by proposing some novel applications of the optimistic security scheme.

2 Optimistic Security

2.1 Overview

The basic approach of an optimistic security system is to assume that any access is legitimate and should be granted. Its goals can be perhaps best summarised by the principle suggested by Bob Blakeley in his 1996 NSPW paper, *The Emperor's Old Armor*[1]:

> Make the users ask forgiveness not permission.

In an optimistic system, enforcement of the security policy is retrospective, and relies on administrators to detect unreasonable access and take steps to compensate for the action. Such steps might include:

- Undoing illegitimate modifications
- Taking punitive action (e.g. firing, or prosecuting individuals)
- Removing privileges.

*The work reported in this paper has been funded in part by the Cooperative Research Centre Program through the Department of Industry, Science and Tourism of the Commonwealth Government of Australia.

These measures should act as both a deterrent and a means to recover the system to a valid state.

Such a system assumes that the risk of failure and the cost of recovery is low compared to the cost of not granting access in a given situation. Optimistic security is not suited to financial or trading systems where the risk of fraud is high, but may be useful in situations such as the protection of private medical information, where emergency access may save someone's life, or in a time-critical system where the person's with the necessary privileges to perform some task may be unavailable.

Optimistic measures may exist alongside a traditional pessimistic scheme, with a mechanism to exceed the current privilege set under certain circumstances. This is similar to existing schemes such as the UNIX setuid system call which allows a user to gain superuser access for certain trusted programs. However, the idea of optimistic security is to provide the ability to rollback from such actions, and to ensure that they are securely audited. The traditional authorisation set will be allocated in line with good security design principles, such as least privilege and separation of duty [6] and will be reserved for everyday use of the system. The organisation may then define either an inclusive or exclusive list of authorisations for use in exceptional circumstances (which may also be explicitly defined). In general the authorisations in the exceptional set will be much less restrictive than the everyday set, and only those actions which may cause catastrophic or irrecoverable damage will be excluded.

2.2 Requirements for Optimistic Security

Providing an optimistic security system requires ways to ensure that the likelihood and consequences of a user maliciously or inadvertedly misusing the system are minimised. To meet this objective the following controls should be considered:

2.2.1 Constrained entry points

Exceeding privilege should be a rarity, rather than a norm. If users need to exceed their privileges on a regular basis, then the organisation should consider whether the list of privileges given to the user is too restrictive, or whether the access control mechanism is too inflexible to support complex policies. It should not be possible for a user to accidentally invoke higher privileges, but should require an explicit, conscious decision. Users should be warned that the mechanism must only be used after careful consideration, and that misuse will have negative (external) consequences. This provides a deterrent, and reminds users of their obligations to the organisation.

The risk of misuse may also be limited by using a threshold scheme to ensure that m of n users must agree to the extra privileges before they will be granted. This limits the risk of a single malicious user causing damage to the system.

2.2.2 Accountability

The system must strongly authenticate users so that they may be associated with given actions. This provides a deterrent, as users know they will be clearly implicated in any misuse, and ensures that only the guilty are targets of any punitive action. If the authentication mechanisms are inadequate, then the risk of misuse will be unacceptable, as users will be able to both masquerade as other users, and repudiate their own illegitimate actions.

2.2.3 Auditability

The system must log the actions of users in detail, so that a postmortem analysis can determine whether an access is legitimate or not. It may be useful to require that users give a reason for using the optimistic mechanisms, and that this is associated with the audit data. The audit data should also be kept secure so that access to the optimistic mechanism does not allow audit information to be compromised. It should be noted that analysis of the audit trail by a system administrator will be labour intensive – a further motivation for educating users to use the mechanism only in extreme circumstances.

In addition to information about the action being logged, other data such as the state of the current system may also be required to determine whether an access was justified.

It should be noted, that an important issue to be addressed will be the retention period for such audit data. This is an important consideration for practical implementations of optimistic security, as the ability to recover from breaches may be limited by the ability to reconstruct an accurate picture of the breach from audit logs.

2.2.4 Recoverability

Accesses which write, modify or delete data must be able to be rolled back to ensure that a user cannot irreparably damage a system. Actions which have external behaviour (e.g. firing a missile, sending a letter) should be associated with compensating actions to restore the system to a stable state (e.g. abort the missile, send apology letter). The issue of recovery in advanced concurrency control systems has been well studied,[1] and as such is not further considered in this paper. In general it is assumed that for any transformation on data or security properties of that data (confidentiality, integrity etc) there is a compensating transaction which exists to reverse this transformation.[2]

2.2.5 Deterrents

One effective way of reducing risks in an optimistic system is by using punitive measures to deter misuse. The punitive measures themselves can be either optimistic – with the system administrator enforcing the measure on detection of misuse; or they can be pessimistic – with the punitive measure implemented immediately and reversed if the action is determined to be legitimate. An example of a workable pessimistic punishment scheme suggested in [1] is to automatically debit a user's bank account when an action is invoked, and refund the money if the access is deemed legitimate.

3 Formal model

To show how integrity can be maintained in an optimistic system, a formal model is needed that ensures the requirements outlined above are realised by the system. The formal model presented in this section is based on the Clark-Wilson Integrity model[2], and supports the notions of accountability, auditability and recoverability.

[1] See IEEE's *Executive Briefing: Advances in Concurrency control and Transaction Processing*[5] for an excellent overview of the state of the art in this area. Of particular interest for optimistic security is the notion of sagas[3], which are long running transactions consisting of independent components. Sagas use compensating transactions to maintain consistency

[2] Where this situation does not exist, it would be inappropriate to use an optimistic system anyway.

Clark and Wilson defined the concept of a *well-formed transaction* as a transaction where the user is unable manipulate data arbitrarily, but only in constrained ways that preserve or ensure the integrity of the data [2]. A security system in which transactions are well-formed ensures that only those actions that have been certified by an administrator as safe can be executed.

Clark and Wilson's formal model for data integrity consists of nine rules for constraining transactions. The rules describe constraints on transformations operating on two types of data:

Constrained Data Items (CDIs) Data items to which the integrity model must be applied.

Unconstrained Data Items (UDIs) Data items not covered by the integrity policy (eg. information typed by the user on the keyboard).

In addition, the Clark Wilson model defines Integrity Verification Procedures (IVPs), which are used to verify that CDIs are in a valid state, and Transformation Procedures (TPs), which are functions that meet the definition of a well-formed transaction. The model uses two types of rules – certification rules enforced by the administrator; and enforcement rules guaranteed by the system. The nine rules are summarised in Figure 1.

Rule	Description
C1	IVPs must be certified to ensure that the system is valid
C2	TPs on CDIs must be certified to ensure that they result in a valid CDI
C3	TPs must be certified to ensure they implement the principles of separation of duties & least privilege
C4	TPs must be certified to ensure that their actions are logged
C5	TPs which act on UDIs must be certified to ensure that they result in a valid CDI
E1	Only certified TPs can operate on CDIs
E2	Users must only access CDIs through TPs for which they are authorised
E3	Users must be authenticated
E4	Only administrator can specify TP authorisations

Figure 1: Clark-Wilson Integrity Rules

3.1 Partially-formed transactions

A *partially-formed transaction* is defined as a transaction where the integrity of the data is not guaranteed, but where a compensating transaction exists to return the system to a valid state. The transaction is only partially-formed, as the integrity of the system is guaranteed by the compensating transaction, and not by constraining the actual action itself.

In this section, a formal model for integrity using partially-formed transactions is described[3]. This model is based on Clark-Wilson's integrity model, but also adds the following components:

PTP A Partial Transformation Procedure. This corresponds to the concept of a partially-formed transaction and describes a procedure which operates on CDIs, but which is not guaranteed to result in valid CDIs.

Compensating TP A transformation procedure which reverses the actions of a PTP.

PCDI A partially-constrained data item. A CDI which has been operated on by a PTP.

Like the Clark Wilson model, IVPs are needed to verify that the system is in a valid state both before and after the execution of one or more partially-formed tranqsactions. This gives the first rule in the integrity model for partially-formed transactions, which is identical to that in the Clark-Wilson model:

C1 IVPs must be certified to ensure that all data items are in a valid state at the time the IVP is run.

The second certification rule in the system applies to PTPs, and outlines the main requirement for optimistic security – i.e. the existence of a valid compensating transaction. This is the cornerstone of the partially-formed transaction integrity model.

C2 All PTPs must be certified to provide a compensating TP that will return any modified CDI to a valid state.

The following enforcement constraints exist to ensure that the PTP is authorised, and that the accountability and auditability requirements are maintained.

E1 The system must ensure that only PTPs that have been certified against requirement C2 are allowed to run.

E2 The system must ensure that users can only use those PTPs for which they have been authorised.

E3 The system must authenticate the identity of each user.

E4 Each PTP must write to an append-only log all the information required to reconstruct and reverse the operation

E5 Only an administrator is permitted to authorise users to access PTPs.

These rules have similar counterparts in the Clark-Wilson model. However, rule E4 is specified as an enforcement rule rather than a certification rule (as in C4 in Clark-Wilson). This is a tacit recognition that programs usually do a poor job of audit, and as the integrity of the model relies on the ability to recognise and reverse anomalous behaviour, we need to ensure this function. This rule requires then that the operating system/security system provide audit *independently* of the programs it is constraining. An example of this is the *tudo* system which uses the system call tracing features of the Solaris /proc filesystem to enable changes to constrained files to be audited and recovered[4].

In addition to the above rules, a mechanism is needed for ensuring that CDIs operating on by PTPs can be validated. This is done by first requiring that PTPs mark CDIs they have accessed as PCDIs, and then by ensuring that rules exist to convert these PCDIs back to valid CDIs. Hence the following rules:

[3] This model has benefited from feedback during the workshop. In particular, rules have been added to account for dependency relationships in the transactions, and the concepts of PTPs and PCDIs have been added to avoid confusion with the Clark-Wilson model. For a description of the rules for partially-formed transactions as they were initially proposed, see [4].

E6 CDIs which are acted on by PTPs must be marked as PCDIs

C3 Compensating TPs must be certified to result in valid CDIs.

C4 The administrator must certify PCDIs as being valid CDIs or if invalid, must apply the compensating transaction to restore the PCDIs to valid CDIs.

Rule C3 is needed to ensure that the compensating transaction is well-formed. In theory, this is a necessary precondition for ensuring integrity. However, in practice it may be possible to relax this constraint (see section 3.1.2). Rule C4 is needed to ensure that the administrator enforces the retrospective security policy.

Lastly, the system needs to deal with the case where PCDIs are used as inputs to other PTPs. In the case that a PTP is reversed, we need to ensure that any other PTPs that rely on the compensated PTP are also reversed. Hence the following rule:

E8 If a PTP on a PCDI is reversed via a compensating TP, then all subsequent PTPs to PCDIs that depend on this item must also be reversed.

Note that the definition of *depend* is left to the implementer. In a system in which TPs consist of simple reads and writes, the *depends* relation could be simply defined by the recursive predicate:

$$\begin{aligned}&a\ \textit{depends}\ b \Rightarrow\\&(\exists \mathrm{PTP}_i \bullet \mathrm{PTP}_i \text{ read } b \wedge \mathrm{PTP}_i \text{ wrote } a) \vee\\&(\exists \mathrm{PTP}_j \bullet \exists c \bullet (\mathrm{PTP}_j \text{ read } c \wedge \mathrm{PTP}_j \text{ wrote } a) \wedge (c\ \textit{depends}\ b))\end{aligned}$$

Together these three certification, and eight enforcement rules constitute the basis of an integrity model for partially-formed transactions. In a manner similar to the Clark-Wilson integrity model, it can be shown that the application of these rules leads to a secure system. To summarise:

- Rule C1 ensures that we can be certain that the system is initially valid.
- Rules C2 and E1 ensure that any transformations to the system can be reversed, and rule C3 ensures that this reversal results in a valid system.
- Rules E2-E4 provide the accountability and auditability properties which are desirable to reduce risks in an optimistic system, and which may be necessary to enforce the compensating TP.
- Rule E5 makes the scheme mandatory.
- Rules E6 and C4 ensure that either the system is verified to be valid, or the compensating transaction is applied. This means that integrity is guaranteed for both those transformations that are legitimate, and those that are deemed to be violations of the security policy.
- Rule E8 guarantees integrity for those transformations which rely on other erroneous transformations.

By iterating through these rules, we can see that if the system is initially valid, the application of a PTP will always lead to a valid system.

3.1.1 Reducing program certification

The partially-formed transaction rules mean that less certification of programs needs to be performed than is required for well-formed transactions. This is possible, as it is generally easier to determine what effects a program has had on its environment than it is to certify it as exhibiting a given behaviour. If integrity is assured after the fact, it is possible to compare actual behaviour with expected behaviour, and reverse the actions in the event of a compromise. Hence, while partially-formed transactions require only one less certification rule, certification of compensating transactions should be much simpler than certifying individual programs as it should be largely possible to provide compensating TPs which are generic for a large range of applications (e.g. reverse all modifications to the filesystem). However, it should be noted that the gain from reduced certification of programs needs to be balanced against the extra load on the security administrator in analysing the audit logs and applying IVPs to determine whether or not the accesses were legitimate. It is interesting to note that Clark and Wilson pointed to the large number of certification rules as a weakness in their model:

> It is desirable to minimise certification rules, because the certification process is complex, prone to error, and must be repeated after each program change. In extending this model, therefore, an important research goal must be to shift as much of the security burden as possible from certification to enforcement[2].

3.1.2 Composition of well-formed and partially-formed transactions

One important issue is whether TPs and PTPs can be composed. This is necessary if optimistic and pessimistic security systems are to coexist. In order to determine this, there are two issues which need to be addressed:

Composibility of completed PTPs the first issue is whether PTPs which have either been validated or compensated can be composed with Clark-Wilson TPs. Providing the TP does not rely on any intermediate state of the PTP, then the integrity properties of the partially-formed transaction model ensure that the TP should operate on valid objects. This idea can be taken further and we can see that taken as a single atomic unit, a sequence of one or more PTPs which have all either been validated or compensated, actually meet the requirements for a Clark-Wilson TP. This is because, providing all the rules in the partially-formed transaction integrity model hold, the C2 rule from the Clark-Wilson model (TPs on CDIs must result in valid CDIs) must also hold. This property allows us to arbitrarily nest PTPs inside TPs. Another corollary alluded to earlier, is that it is possible for a compensating transactions to contain elements which are partially-formed but which when taken as a single atomic unit, form a well-formed transactions. This can provide the equivalent of local undo-redo semantics, where some elements of a compensating transaction also have the ability to be recovered from. The idea of encapsulating a number of transactions which are not well-formed within a TP is consistent with the Clark-Wilson model, as Clark and Wilson themselves state:

> During the mid-point of the execution of a TP, there is no requirement that the system be in a valid state[2].

While Clark and Wilson were referring here specifically to serialiseability and concurrency control of TPs, the same applies to any transaction which exhibits external consistency, but which may be internally inconsistent.

Composibility of TPs with PCDIs in some cases it may be desirable to have TPs which operate on PCDIs, thus exposing the intermediate state of the PTP. A concrete example is the ability for the compensating action to be well-formed. However, this issue is simply addressed by having the TP treat the PCDI as though it were a UDI. In this case, the Clark-Wilson certification rule C5 (TPs which act on UDIs must be certified to ensure they result in a valid CDI), provides the necessary properties for these procedures to be composed.

4 Applications of optimistic security

4.1 Emergency "break-glass" tools

Equipment such as alarms, emergency stop buttons and fire-fighting equipment (e.g. axes, hoses etc) are often stored in "break-glass" containers for emergency use. The benefit of such containers is they require users to make a conscious decision about using the equipment, thus preventing accidental misuse. In addition, "break-glass" containers are usually labelled with warnings about the penalties of deliberate misuse, providing a significant deterrent.

Such a device in a computer security system could be a useful in coping with an emergency situation. The software equivalent of the "break-glass" container would be a program which is suitably constrained using an optimistic security system, and which gives stern warnings about misuse before it is activated. In an emergency situation, the user could operate the tool, but would have to explain themselves to the system administrator after the event in order to avoid the associated penalty. If the system administrator decided the use was not legitimate, they could use the recovery mechanism and enforce the penalty.

4.2 Retrospective content filtering

One of the negative aspects of systems which provide filtering of material which is deemed harmful or inappropriate, is that the algorithms used to determine which content to filter can often result in false matches. The result of this is that users can be denied access to legitimate content, leading them to search for ways to circumvent the system.

By applying the principles of optimistic security, users would be able to access any material they desired, and an administrator would log all material accessed and run the content filtering algorithm retrospectively. This would give a list of matches which the administrator could then further investigate to determine whether the content is inappropriate.

Such a mechanism when coupled with an appropriate system of punitive measures (e.g. reprimanding a child, dismissing an employee, or posting a list of those users who accessed pornography in the last week on the company notice-board!) can be more effective than simply disallowing access as its enforcement enables the administrator to reinforce the policy to both the culprit and any other potential perpetrators.

4.3 Sandboxing "somewhat-trusted" applications

Traditionally, the focus of "sandboxing" (or constraining the access privileges of programs) has been on untrusted code that is downloaded from the Internet. However, many users of personal operating systems use a large number of applications which have unrestrained access, and hence the potential to damage the integrity of their systems. For such applications, it is often not possible to maintain integrity using traditional Clark-Wilson principles, as it is either too expensive to certify the program, or its source code is unavailable for certification.

By creating a sandbox along optimistic principles, the damage caused by the use and misuse of these "somewhat-trusted" applications could be limited, while still allowing full-functionality. For example: an optimistic sandbox could track the changes made to the filesystem by a word processing program, and allow the user to undo these changes in the event of a crash or malicious macro virus. This would result in greatly improved security (and safety) for these applications without loss of functionality, or expensive certification of the programs.

4.4 Watching your system administrator watching you

The formal integrity model presented in this paper specifies a mandatory access control system (authorisation determined by an administrator). This is an artifact of basing the model on Clark-Wilson. However, there is no reason why an optimistic security model could not be applied to a discretionary access control system (authorisation determined by the objects owner/creator). Currently system administrators have more or less unfettered access to a system including data which users may wish to keep private (e.g. private email). By applying an optimistic security system, access to these files could be constrained such that the user is informed whenever an administrator accesses files that they consider to part of their private set. There may be occasions where such access is legitimate, but others where this may be intrusive or a breach of the users privacy. By ensuring that users know when management/administrators are accessing their files, users can have more confidence that their privacy is being maintained.

5 Conclusions

The Clark-Wilson integrity model provides a means by which a system can be constrained to ensure that only legitimate accesses can be executed. However, this paper argues that under exceptional circumstances this requirement should be able to be relaxed. The notion of a partially-formed transaction provides a mechanism by which a system can seek to be optimistic about authorising user actions, while still maintaining system integrity. It is believed that such a mechanism can help to increase the flexibility of security systems in environments where hard and fast rules are not always the best option.

This paper has shown that an integrity model for an optimistic system is feasible, and moreover that providing certain preconditions, the transformations for such a system are composable with those which use a more traditional pessimistic system. Finally a number of novel applications of the optimistic security model are given that show how such a system could be useful.

References

[1] Bob Blakeley. The emperor's old armor. In *Proceedings of the 1996 New Security Paradigms Workshop*, pages 2–16. ACM, September 1996.

[2] David D. Clark and David R. Wilson. A comparison of commercial and military security policies. In *1987 IEEE Symposium on Security and Privacy*, pages 184–194, Oakland, CA, April 1987.

[3] H. Garcia-Molina and K. Salem. Sagas. In *Proceedings of the ACM SIGMOD International Conference on the Management of Data*, pages 249–259, New York, NY, 1987. ACM press.

[4] Dean Povey. Enforcing well-formed and partially-formed transactions for UNIX. In *Proceedings of the 8th USENIX Security Symposium*. USENIX Association, August 1999.

[5] Krithi Ramamritham and Panos K. Chrysanthis. *Executive Briefing: Advances in Concurrency Control and Transaction Processing*. IEEE Computer Society Press, Los Alamitos, CA, 1997.

[6] Jerome H. Saltzer and Michael D. Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63(9):1278–1308, September 1975.