Proc. of the 15th NIST-NCSC National Computer Security Conference, Baltimor e, MD, October 1992, pages 221-235

A LATTICE INTERPRETATION OF THE CHINESE WALL POLICY

Ravi S. Sandhu[1]

Center for Secure Information Systems
&
Department of Information and Software Systems Engineering
George Mason University, Fairfax, Virginia 22030

Abstract The Chinese Wall policy was identified and so named by Brewer and Nash [2]. This policy arises in the segment of the commercial sector which provides consulting services to other companies. Consultants naturally have to deal with confidential company information for their clients. The objective of the Chinese Wall policy is to prevent information flows which cause conflict of interest for individual consultants. Brewer and Nash develop a mathematical model of the Chinese Wall policy, on the basis of which they claim that this policy "cannot be correctly represented by a Bell-LaPadula model." In this paper we demonstrate that the Brewer-Nash model is too restrictive to be employed in a practical system. This is due to their treatment of users and subjects as synonymous concepts, with the consequence that they do not distinguish security policy as applied to human users versus security policy as applied to computer subjects. By maintaining a careful distinction between users, principals and subjects, we show that the Chinese Wall policy is just another lattice-based information policy which can be easily represented within the Bell-LaPadula framework.

1 INTRODUCTION

The Chinese Wall policy arises in the segment of the commercial sector which provides consulting services to other companies. The policy was identified by Brewer and Nash [2]. It attracted considerable interest in the security community, because it is a real-world information flow policy in the commercial sector rather than the usual military or government sectors. Moreover, it has characteristics which are quite different from the military security policy considered in the Bell-LaPadula model [1].

The objective of the Chinese Wall policy is to prevent information flows which cause conflict of interest for individual consultants. Consultants naturally have to deal with confidential company information for their clients. A single consultant should not have access to information about two banks or information about two oil companies, etc., because such insider information creates conflict of interest in the consultant's analysis and disservice to the clients. Insider information about companies of the same type also presents the potential for consultants to abuse such knowledge for personal profit.

The Chinese Wall policy has a dynamic aspect to it. Consider a consultant who is new in the field, say fresh out of Graduate School. At this point there is no mandatory restriction on the consultant's

[1] This work was partially supported by the National Security Agency through contract MDA904-92-C-5141.

access rights. The consultant can access information about any company in the database (restricted only by discretionary controls which we will be ignoring throughout this paper). Now say that the consultant accesses information about bank A. Thereafter that consultant is mandatorily denied access to information about any other bank. There are, however, still no mandatory restrictions regarding that consultant's access to oil companies, insurance companies, etc.

Largely due to this dynamic aspect, Brewer and Nash claim that the Chinese Wall policy "cannot be correctly represented by a Bell-LaPadula model." One objective of our paper is to dispute this claim, by showing how the Chinese Wall policy is just another example of a lattice-based information flow policy which can be easily represented within the Bell-LaPadula framework.[2]

Another objective of our paper is to show the vital importance of distinguishing security policy as applied to human users versus security policy as applied to computer subjects. Brewer and Nash fail to make this distinction. They treat users and subjects as synonymous concepts. As a result their model is much too restrictive to be employed in a practical system. By maintaining a careful distinction between users, principals and subjects, we develop a model for the Chinese Wall policy which addresses threats from Trojan Horse infected programs. The Brewer-Nash model on the other hand makes a futile attempt to safeguard against malicious consultants.

The rest of this paper is organized as follows. Section 2 reviews the distinction between users, principals and subjects in a computer system. Section 3 discusses the Chinese Wall policy and the threats that it addresses. We carefully distinguish between threats posed by malicious consultants versus threats posed by Trojan Horse infected programs. While computer security can address threats posed by Trojan Horse infected programs, it cannot fully address threats posed by malicious consultants. After all, consultants who choose to share information in violation of Chinese Walls can do so equally efficiently by communication means outside of the computer system. With this context we analyze the Brewer-Nash model in section 4 and show that this model is unduly restrictive. In section 5 we develop a lattice-based model for the Chinese Wall policy and relate it to the Bell-LaPadula model. Section 6 concludes the paper.

2 USERS, PRINCIPALS AND SUBJECTS

To understand the Chinese Wall policy and its nuances with respect to subjects versus human users, we must first understand the distinction between *users*, *principals* and *subjects*. This distinction is fundamental to computer security and goes back to the beginnings of the discipline. Nevertheless, it is often dealt with imprecisely in the literature leading to undue confusion about the objectives of computer security.

2.1 Users

We understand a user to be a human being. We assume that each human being known to the system is recognized as a unique user. In other words the unique human being Jane Doe cannot have more than one user identity in the system. If Jane Doe is not an authorized user of the system she has no user identity. Conversely, if she is an authorized user she is known by exactly one user identity, say, JDoe. Clearly this assumption can be enforced only by adequate administrative controls, which we assume are in place. It should be noted that violation of this requirement is often the cause of security violations in current systems.

[2] In fairness to Brewer and Nash it should be noted that the original Bell-LaPadula model is inadequate to express the Chinese Wall policy. The model given here does require (i) a careful distinction between users, principals and subjects, and (ii) the concept of user labels which "float up" versus labels on principals, subjects and objects which do not change.

2.2 Principals

Our concept of principal is adapted from Saltzer and Schroeder [6]. Each user may have several principals associated with the user. On the other hand each principal is required to be associated with a single user.

The motivation in [6] for this concept was that different principals would correspond to, say, different projects on which the user works. Every time a user logs in to the system it is as a particular principal. Thus if Jane Doe was assigned to projects Red and Blue, she would have three principals associated with her user identity, say, JDoe, JDoe.Red and JDoe.Blue. On any session Jane could login as any one of these principals, depending on the work she planned to do in that session. Each principal associated with JDoe obtains a different set of access rights. Thus JDoe.Red has access to the files and other objects of project Red, but not project Blue. Similarly, JDoe.Blue has access to the files and other objects of project Blue, but not project Red. The principal JDoe is a generic principal for Jane allowing access to her personal files, but not to any of the project files.

The notion of principal reflects the everyday reality that individuals wear several different "hats" in an organization, with their authority and responsibility determined by the particular "hat" they are wearing at a given moment. Saltzer and Schroeder introduce principals in a discretionary context. The concept carries over equally well to mandatory policies. We often encounter phrases such as, "the top-secret user John logs in at the secret level," in the security literature. What are we to make of this statement? In the user-principal terminology we interpret this statement as follows:

- Firstly, there is a unique user John, cleared to top-secret, independent of the level at which John logs in.
- Secondly, John can log in at every level dominated by top-secret. At each of these levels there is a separate principal associated with John. So John.top-secret is the principal when John logs in at top-secret, John.secret is the principal when John logs in at secret, etc.

We will see that this concept of a principal is the key to achieving lattice-based enforcement of Chinese Walls.

2.3 Subjects

We understand a subject to be a process in the system, i.e., a subject is a program in execution. Each subject is associated with a single principal on behalf of whom the subject executes. In general a principal may have many subjects associated with it concurrently running in the system.

For simplicity we assume that a subject executes with all the privileges of its associated principal.[3] Thus when Jane Doe logs in as JDoe.Red and invokes her favorite editor Emacs, a subject associated with JDoe.Red is created and runs the Emacs code. This subject acquires all the access rights of the principal JDoe.Red. Similarly when John logs in as John.top-secret every subject spawned during that session runs at the top-secret level.

To summarize

- each authorized human user is known as a unique user to the system,[4]

[3] This is the actual situation in most existing systems, including those specifically designed for security. More generally a subject could be created with a proper subset of privileges of its associated principal. The most general case is to allow a subject to have multiple parents, from each of whom it obtains some privileges.

[4] This requirement is admittedly violated in many systems, and will require administrative controls outside of the computer system. Nevertheless, without this requirement there is little scope for enforcing aggregation policies such as Chinese Walls. Moreover, it is also a prerequisite for enforcing separation of duties.

Company Information

Conflict of Interest Class i ... Conflict of Interest Class j

Company i.1 ... Company i.m ... Company j.1 ... Company j.n

Figure 1: Company Information in the Chinese Wall Policy

- each user can log in as one of several principals but each principal is associated with only one user, and
- each principal can spawn several subjects but each subject is associated with only one principal.

3 THE CHINESE WALL POLICY

The Chinese Wall policy is intuitively simple and easy to describe. In this section we describe this policy by adapting the description of Brewer and Nash [2] and adding additional concepts to it. It is important to keep in mind that we are deliberately ignoring all discretionary access control issues in this paper. In practice the Chinese Wall policy as described here would be the mandatory component of a larger policy which includes additional discretionary controls (and possibly additional mandatory controls).

We begin by distinguishing *public* information from *company* information. There are no mandatory controls on reading public information. Reading company information on the other hand is subjected to mandatory controls, which we will discuss in a moment. The policy for writing public or company information is derived from its consequence on providing possible indirect read access contrary to the mandatory read controls. It is in this respect that users and subjects must be treated differently. We will consider mandatory controls on writing information following our discussion of the read controls.

The motivation for recognizing public information is that a computer system used for consulting services will inevitably have large public databases for use by consultants. Moreover, public information allows for desirable features such as public bulletin boards and electronic mail which users expect to be available in any modern computer system. Public information can be read by all users, principals and subjects in the system (restricted only by discretionary controls which, as we have said, we are ignoring in this paper).

Company information is categorized into mutually disjoint conflict of interest classes as shown in figure 1. Each company belongs to exactly one conflict of interest (COI) class. The Chinese Wall policy requires that a consultant should not be able to read information for more than one company in any given COI class. To be concrete let us say that COI class i consists of banks and COI class j consists of oil companies. The Chinese Wall stipulation is that the same consultant should not have read access to two or more banks or two or more oil companies.

The Chinese Wall policy has a mix of free choice and mandated restrictions. So long as a consultant has not yet been exposed to any company information about banks, that consultant has the potential to read information about any bank. The moment this consultant reads, say, bank A information, thereafter that consultant is to be denied read access to all other banks. The free choice of selecting the first company to read in a COI class can be exercised once and is then forever gone (or at least gone for a sufficient length of time to avoid conflict of interest).

So long as we have focussed on read access the Chinese Wall policy has been easy to state and understand. When we turn to write access the situation becomes more complicated and subtle. This is the usual case with confidentiality policies. For example, the simple-security rule of the well-known Bell-LaPadula model [1] is similarly intuitive and straightforward whereas the $\star$-property (which prohibits write down) is more subtle.

In computer security it is easy to confuse the threat from malicious users with the threat from malicious subjects. In the Bell-LaPadula model, mandatory controls on write access are imposed to prevent Trojan Horse infected subjects from leaking information contrary to the system policy. These controls do not address the threat of malicious human users. It should always be kept in mind that a malicious user can compromise information confidentiality by employing communication means outside of the computer system. Thus John as a human being cleared to top-secret is nevertheless able to write and publish unclassified documents. This is because John is trusted not to leak top-secret information in his unclassified writings. On the other hand malicious subjects executing with John's top-secret privileges can leak top-secret information if not constrained by the $\star$-property.

In much the same way a computer system cannot solve the problem of a malicious consultant. A determined consultant can leak damaging confidential information about a company to, say, the Wall Street Journal by means of a telephone call. Similarly, a consultant can provide insider company information directly to its competitors or share this information with other consultants. Just as our top-secret user John is trusted not to divulge secrets, so must our consultants be trusted as individuals not to break Chinese Walls.

4 THE BREWER-NASH MODEL

We now consider the Brewer-Nash model for the Chinese Wall policy. In this model data is viewed as consisting of *objects* each of which belongs to a *company dataset*. The company datasets are categorized into conflict of interest (COI) classes, along the lines of 1.

The Brewer-Nash model does not distinguish users, principals and subjects. It uses the single concept of subject for all three notions. This leads them to propose the following mandatory rules.

1. *BN Read Rule*: Subject S can read object O only if
   - O is in the same company dataset as some object previously read by S (i.e., O is within the wall), or
   - O belongs to a COI class within which S has not read any object (i.e., O is outside the wall).
2. *BN Write Rule*: Subject S can write object O only if
   - S can read O by the BN read rule, and

- no object can be read which is in a different company dataset to the one for which write access is requested.

We have called these the BN read rule and BN write rule for ease of reference. They are analogous to the simple-security and $\star$-properties of the Bell-LaPadula model.

The BN read rule conveys the dynamic aspect of the Chinese Wall policy. This rule clearly applies to the human users, viz., the consultants, in the system. Since the Brewer-Nash model does not distinguish between users and subjects, this rule is also applied to all subjects in the system.

The BN write rule is brought in to prevent Trojan Horse laden subjects from breaching the Chinese Walls. To see its motivation consider that consultant John has read access to Bank A objects and Oil Company OC objects, and that consultant Jane has read access to Bank B objects and Oil Company OC objects. Individually John and Jane are in compliance with the Chinese Wall policy. Now suppose John is allowed write access to OC objects. A Trojan Horse infected subject running with John's privileges can thereby transfer information from Bank A objects to OC objects. These OC objects can be read by subjects running on behalf of Jane, who then has read access to information about Bank A and Bank B.[5]

The BN write rule is successful in preventing such information leakage by Trojan Horses. However, it does so at an unacceptable cost. It is easy to see that the BN write rule has the following implication.

- A subject which has read objects from two or more company datasets cannot write at all.
- A subject which has read objects from exactly one company dataset can write to that dataset.

These implications are clearly unacceptable (if the computer system is to be used for something more than a read-only repository of confidential information). Under this regime a consultant can work effectively so long as he or she is assigned to exactly one company. The moment the consultant is assigned to a second company, he or she will be unable to write *any* information into the system.

Fortunately these implications are not inherent in the Chinese Wall policy. They are rather a consequence of the Brewer-Nash model's failure to distinguish rules applied to users from rules applied to subjects. The key observation is that we can live with the implications listed above with respect to subjects, but not with respect to users. In particular, limiting every subject to reading and writing a single company dataset is an acceptable restriction. Thus, any subject executing on behalf of John should either be able to read and write Bank A objects, or read and write Oil Company OC objects. John as a human being is, however allowed to read and write both Bank A and Oil Company OC objects. For that matter, John is also allowed to read and write public objects. However, he is not allowed to do all of these actions using the same subject.

5 A LATTICE INTERPRETATION

In this section we provide a lattice-based interpretation of the Chinese Wall policy. It was shown by Denning [3] that information flow policies in general require that objects be labeled with a lattice structure. Denning's result is derived from the following axioms.

1. Information flow is reflexive, transitive and symmetric.
2. There is a lowest class of information which is allowed to flow into all other classes.

[5] Note that Computer Security cannot do anything to prevent John and Jane from exchanging Bank A and Bank B information outside of the computer system. But in such an exchange John and Jane are accomplices. In the example given here John is not an accomplice but rather an unwitting victim of a Trojan Horse.

3. For any two classes of information A and B there is a class C which is the least upper bound of A and B (i.e., (i) information from both A and B can flow to C, and (ii) for all classes D such that information can flow from both A and B it is the case that information can flow from C to D).

These axioms are generally accepted as being very reasonable.[6] Now there is nothing in the Chinese Wall policy that is contrary to these axioms. We will bear out this claim by showing how we can construct a lattice structure for the Chinese Wall policy. We do so by defining a number of axioms below.

5.1 The Lattice Structure for Chinese Walls

Let us begin by introducing the conflict of interest classes and companies.

A1. There are n conflict of interest classes: COI_1, COI_2, $\ldots$, COI_n.

A2. $COI_i = \{1, 2, \ldots, m_i\}$, for $i = 1, 2, \ldots n$, i.e., each conflict of interest class COI_i consists of m_i companies.

In other words there are n conflict of interest classes, each of which contains some number of companies as visually depicted in figure 1.

We propose to label each object in the system with the companies from which it contains information. Thus an object which contains information from Bank A and Oil Company OC is labeled {Bank A, Oil Company OC}. Labels such as {Bank A, Bank B, Oil Company OC} are clearly contrary to the Chinese Wall policy. We prohibit such labels in our system by defining a security label as an n-element vector $[i_1, i_2, \ldots, i_n]$, where each $i_k \in COI_k$ or $i_k = \perp$.

An object labeled $[i_1, i_2, \ldots, i_n]$ is interpreted as signifying that it contains information from company i_1 of COI_1, company i_2 of COI_2 and so on. When an element of the vector is $\perp$ rather than an integer, it means that the object has no information from any company in the corresponding conflict of interest class. For example, an object which contains information only from company 4 in COI_3 will be labeled with the vector $[\perp, \perp, 4, \perp, \ldots, \perp]$, i.e., all elements other than the third one will be $\perp$. Similarly, an object which contains information from company 7 in COI_2 and company 5 in COI_4 will be labeled with the vector $[\perp, 7, \perp, 5, \perp, \ldots, \perp]$.

This leads us to the following definition for the set of labels.

A3. $LABELS = \{[i_1, i_2, \ldots, i_n] | i_1 \in COI'_1, i_2 \in COI'_2, \ldots, i_n \in COI'_n\}$ where $COI'_i = COI_i \cup \{\perp\}$

Note that the label which has all $\perp$ elements naturally corresponds to public information. There is, however, no naturally occurring system high label (in fact such a label is contrary to the Chinese Wall policy). In order to complete the lattice we introduce a special label for system high (which we will not assign to any subject in the system).

A4. $EXTLABELS = LABELS \cup \{SYSHIGH\}$

Next we define the dominance relation among labels as follows, where the notation $l_1[i_k]$ denotes the i_k-th element of label l_1.

A5. $(\forall l_1, l_2 \in LABELS)[l_1 \geq l_2 \Leftrightarrow (\forall i_k = 1, \ldots, n)[l_1[i_k] = l_2[i_k] \vee l_2[i_k] = \perp]]$

[6] Some researchers have tried to relax them further, for instance by dropping the transitive requirement on information flow, but in the main the security community has accepted these.

In other words, l_1 dominates l_2 provided that l_1 and l_2 agree wherever $l_2 \neq \perp$. For example $[1, 3, 2] \geq [1, 3, \perp]$, $[1, 3, 1] \geq [\perp, \perp, 1]$ while $[1, 3, 2]$ and $[1, 2, 3]$ are incomparable. Note that every label dominates the system low label which consists of all $\perp$ elements. To account for the special system high label we have the following axiom.

A6. $(\forall l \in EXTLABELS)[SYSHIGH \geq l]$

To complete the lattice structure it remains to define the least upper bound operator. In order to do so we introduce the following notion.

A7. $l_1, l_2 \in LABELS$ are compatible if and only if for all $k = 1, \ldots, n$, $l_1[i_k] = l_2[i_k] \vee l_1[i_k] = \perp \vee l_2[i_k] = \perp$

In other words, two label are compatible if wherever they disagree at least one of them is $\perp$. Note that if $l_1 \geq l_2$ then l_1 and l_2 are compatible. Labels which are incomparable with respect to the dominance relation may or may not compatible, e.g., $[1, 3, 2]$ and $[1, 2, 3]$ are incompatible while $[1, \perp, 2]$ and $[1, 2, \perp]$ are compatible.

Incompatible labels cannot be legitimately combined under the Chinese Wall policy. This is expressed by the following axiom.

A8. If l_1 is incompatible with l_2 then $lub(l_1, l_2) = SYSHIGH$

For compatible labels the least upper bound is computed as follows.

A9. if l_1 is compatible with l_2 then $lub(l_1, l_2) = l_3$ where $l_3[i_k] = \begin{cases} l_1[i_k] & \text{if } l_i[i_k] \neq \perp \\ l_2[i_k] & \text{otherwise} \end{cases}$

For example, the least upper bound of $[1, \perp, 2]$ and $[1, 2, \perp]$ is $[1, 2, 2]$. Finally to complete the definition with respect to the special system high label, we have the following axiom

A10. $(\forall l \in EXTLABELS)[lub(SYSHIGH, l) = SYSHIGH]$

It is easy to verify that the axioms **A1** to **A10** define a lattice on the set of labels $EXTLABELS$ with dominance relation $\geq$. Information flow occurs in the direction opposite to the dominance relation and is obviously reflexive, transitive and symmetric. The required system low class is identified by the label consisting of all $\perp$ elements, and the least upper bound operator has been defined.

Figure 2 shows a lattice with two conflict of interest classes, each with two companies in it. The lattice is shown by its Hasse diagram, in which the dominance relation goes from top to bottom with transitive and reflexive edges omitted.

5.2 Chinese Wall Model

Given this lattice structure we have developed, let us see how we can solve the Chinese Wall problem. To be concrete we describe our solution in terms of the specific lattice of figure 2. The solution is, however, completely general and applies to any size Chinese Wall lattice.

We require every object in the system to be labeled by one of the labels in figure 2. Public objects are labeled $[\perp, \perp]$. Objects with company information from a single company are labeled as follows:

- $[1, \perp]$: objects with information for company 1 in COI_1.

$SYSHIGH$

$[1, 1]$ $[1, 2]$ $[2, 1]$ $[2, 2]$

$[1, \perp]$ $[2, \perp]$ $[\perp, 1]$ $[\perp, 2]$

$[\perp, \perp]$

Figure 2: Example of a Chinese Wall Lattice

- $[2, \perp]$: objects with information for company 2 in COI_1.
- $[\perp, 1]$: objects with information for company 1 in COI_2.
- $[\perp, 2]$: objects with information for company 2 in COI_2.

Objects with company information from more than one company (without violation of Chinese Walls) are labeled as follows:

- $[1, 1]$: objects with information for company 1 in COI_1 and company 1 in COI_2.
- $[1, 2]$: objects with information for company 1 in COI_1 and company 2 in COI_2.
- $[2, 1]$: objects with information for company 2 in COI_1 and company 1 in COI_2.
- $[2, 2]$: objects with information for company 2 in COI_2 and company 2 in COI_2.

Objects labeled $SYSHIGH$ violate the Chinese Wall policy, in that they can combine information from any subset of the companies. These objects are inaccessible in the system (and therefore might as well not exist).

Now let us consider labels on users, principals and subjects. We treat the label of a user as a high-water mark which can float up in the lattice but not down. A newly enrolled user in the system is assigned the label $[\perp, \perp]$.[7] As the user reads various company information the user's label floats up in the lattice.[8] For example, by reading information about company 1 in conflict of interest class

[7] This assumes that the user is entering the system with a "clean slate." A user who has had prior exposure to company information in some other system should enter with an appropriate label reflecting the extent of this prior exposure.

[8] The exact manner in which a user's label is allowed to float up is an issue of implementation. If the users have complete freedom in this respect, the proposed read access could be specified at the time of login. The system could

1 the user's label is modified to $[1, \perp]$. Reading information about company 2 in conflict of interest class 2 further modifies the user's label to $[1, 2]$.

This floating up of a user's label is allowed, so long as the label does not float up to $SYSHIGH$. Operations which would force the user's label to $SYSHIGH$ are thereby prohibited. The ability to float a user's label upwards[9] addresses the dynamic requirement of the Chinese Wall policy. The floating label keeps track of a user's read operations in the system.

With each user we associate a set of principals, one at each label dominated by the user's label. Thus, if Jane as a user has the label $[1, 1]$, she has the following principals associated with her: Jane.$[1, 1]$, Jane.$[1, \perp]$, Jane.$[\perp, 1]$ and Jane.$[\perp, \perp]$. Each of these corresponds to the label with which she wishes to log in on a given session. These principals have fixed labels which do not change. The floating up of a user's label corresponds to creation of one or more new principals for that user. For example, when Jane had the label $[1, \perp]$, she had only two principals associated with her, viz., Jane.$[1, \perp]$ and Jane.$[\perp, \perp]$. When Jane's label floated up to $[1, 1]$, she acquired two new principals Jane.$[1, 1]$ and Jane.$[\perp, 1]$. This floating up of Jane's label is achieved by Jane's directive to the system. The system will allow this action only if the float up is to some label strictly below $SYSHIGH$.

Each principal has a fixed label. Every subject created by that principal inherits that label. Thus, all activity in the system initiated by Jane.$[1, \perp]$ will be carried out by subjects with the label $[1, \perp]$. The label of a subject is determined by the label of the principal who creates that subject. A subject's label remains fixed for the life of that subject.

All read and write operations in the system are carried out by subjects. These subjects are constrained by the familiar simple-security and $\star$-properties of the Bell-LaPadula model. That is a subject can only read objects whose labels are dominated by the subject's label, and can only write objects whose labels dominate the subject's label.

Now suppose that Jane logs in as the principal $[1, \perp]$. All subjects created during that session will inherit the label $[1, \perp]$. This will allow these subjects to read public objects labeled $[\perp, \perp]$, to read and write company objects labeled $[1, \perp]$, and write[10] objects with labels $[1, 1]$, $[1, 2]$ and $SYSHIGH$.

6 CONCLUSION

In this paper we have given a lattice interpretation of the Chinese Wall policy of Brewer and Nash [2]. In doing so we have disputed[11] their claim that the Chinese Wall policy "cannot be correctly represented by a Bell-LaPadula model." We have also shown that the Brewer-Nash model is too restrictive to be employed in practice, since it essentially prohibits consultants from adding new information into the system. By maintaining a careful distinction between users, principals and subjects, we developed a model for the Chinese Wall policy which addresses threats from Trojan Horse infected programs and retains the ability of consultants to write information into the company datasets they are analyzing. Our paper demonstrates the vital importance of distinguishing security policy as applied to human users versus security policy as applied to computer subjects.

The lattice model we have developed for the Chinese Wall policy uses the Bell-LaPadula simple-

then create a suitable principal for that user session. On the other hand one might constrain this by discretionary access controls which we have ignored in this paper. For instance, a user may be allowed to read only that company information which the user's boss assigns him or her to. In this case the float up of a user's label is effectively done by some other user. Full consideration of such discretionary policies and their interplay with the mandatory policy, would require a model such as the Typed Access Matrix [7].

[9] This float upwards does not present the security problems with changing labels discussed in [5]. This is due to the upward floating or high-water mark nature of our user labels.

[10] As is often done in multilevel secure database systems, we can prohibit this "write up" if we so choose.

[11] Although, see footnote 2 earlier in the paper.

security and $\star$-properties. In this sense it is consistent with the Orange Book [4]. However, the structure of our security labels departs from the conventional military and government sector (with their hierarchical and non-hierarchical components). A system built to Orange Book criteria can be used to enforce Chinese Walls, provided there is some flexibility in the structure of the labels in the system.

References

[1] Bell, D.E. and LaPadula, L.J. "Secure Computer Systems: Unified Exposition and Multics Interpretation." MTR-2997, Mitre, Bedford, Massachusetts (1975).

[2] Brewer, D.F.C and Nash, M.J. "The Chinese Wall Security Policy." *IEEE Symposium on Security and Privacy*, 215-228 (1989).

[3] Denning, D.E. "A Lattice Model of Secure Information Flow." *Communications of ACM* 19(5):236-243 (1976).

[4] Department of Defense National Computer Security Center. *Department of Defense Trusted Computer Systems Evaluation Criteria.* DoD 5200.28-STD, (1985).

[5] McLean, J. "Reasoning About Security Models." *IEEE Symposium on Security and Privacy*, 123-131 (1987).

[6] Saltzer, J.H. and Schroeder, M.D. "The Protection of Information in Computer Systems." *Proceedings of IEEE* 63(9):1278-1308 (1975).

[7] Sandhu, R.S. "The Typed Access Matrix Model." *Proc. IEEE Symposium on Research in Security and Privacy*, Oakland, California, May 1992, pages 122-136.