

# FreeBSD Mandatory Access Control Usage for Implementing Enterprise Security Policies

Kirill Bolshakov  
kirill.bolshakov@ieee.org  
Saint-Petersburg State University of Aerospace Instrumentation

Elena Reshetova  
elena.reshetova@gmail.com

## Abstract

FreeBSD was one of the first widely deployed free operating systems to provide mandatory access control. It supports a number of classic MAC models. This tutorial paper addresses exploiting this implementation to enforce typical enterprise security policies of varying complexities.

## I. INTRODUCTION

Security needs of organizations are becoming more and more sophisticated nowadays. Most general-purpose operating systems (GPOS) provide access control policies to meet these needs. There are cases when the traditionally deployed Discretionary Access Control (DAC) rules are not sufficient: they tend to quickly become unmanageable in the case of large installations, and also are not enough for controlling information flows. This is when the Mandatory Access Control (MAC) comes in: it provides for better manageability and directly targets the information flows. In their turn, the information flows address the confidentiality and integrity needs of information security within an organization. Until very recently, the GPOSes tended to provide various flavors of DAC only. The FreeBSD OS [1] was one of the first widely deployed open source GPOSes to support MAC [2], [3]. In this paper, a number of organizational policy examples are implemented in the environment of the FreeBSD MAC.

The authors strongly believe that in order to implement a sound MAC policy it is important to understand MAC's mathematical foundations. These foundations were set by Denning in [4]. There also exists a terminology confusion between MAC and LBAC (lattice-based access control). These models are the same, because MAC security labels [5] directly correspond to security classes of lattice-based models (this was also pointed to by Sandhu [6] and Osborn [7]).

Let us first address the definition of the information flow. According to Denning and Sandhu, the security policies regulate how the information “flows from one object to another”. A typical object is a shared memory segment, a file system object or a network packet. Obviously, controlling the information flows is important to prevent the leakage of the confidential information, the one usually sought by insiders. Another goal is the forgery prevention, so that no untrusted reports are ever submitted to the top level of the organization hierarchy, and no top-ranking company officers take any unchecked or untrusted information into account during decision making.

To implement the information flow control, every object is assigned a security label (also called a security classification), implemented by the FreeBSD file label. When the information flows from one object into another, *an information flow* from the security class of the first object to the security class of the second one also takes place. Whether the information flow is allowed is regulated by the relation between the object security classes. The subjects are the entities performing the information transfer between the objects. In our case, a subject appears when a user logs in to the system and is assigned a set of privileges. As we are considering MAC, the set of privileges is rendered as security clearance. It is implemented by the FreeBSD user label.

This paper is organized as follows. In the next section an example of an organization and its document flow is described. The following sections implement organization's information security goals, which gradually increase in complexity. The information security goals specify the target effect: preserving data and process integrity, restricting access to the confidential information, or implementing a consulting services policy. For every security goal, a corresponding classic MAC model or a combination of them is chosen. The models are then implemented in the FreeBSD MAC framework.

## II. THE EXAMPLE

The example organization we consider is a very small technology company. There are only six positions within the organization. The organization chart and the exchange of files corresponding to the information flows are shown in Fig. 1. The document flow is bidirectional.

The file system folder layout reflects this, and the folders are listed in Table I. Operations will be applied to the files in these folders. The *Temp* folder is used for file exchange purposes. The administrator may consider configuring this folder with the “sticky” attribute (so that the files could be deleted by their owners only). The “U” (from “Untrusted”) folders are used by Mary, Alice and Robert for delivering documents to Jane and Paul, respectively. After the managers review these untrusted documents and deem them trustworthy, the files may be copied to the corresponding no-prefix folders while increasing their

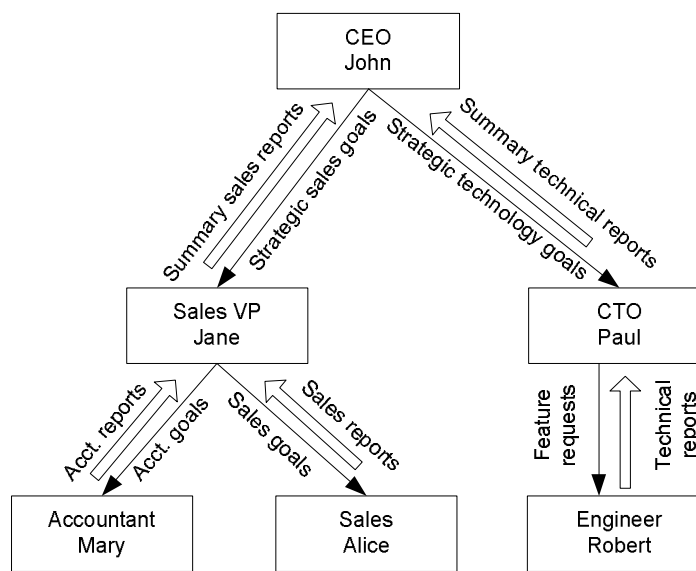


Figure 1. Example Organization Chart and Document Flows

security class. This makes these documents available for further processing to the higher levels of hierarchy: for example, they may be included into Jane's or Paul's reports for John.

In FreeBSD, the MAC labels are assigned to the user accounts via the notion of *login classes*. The mapping of user accounts to their login classes is given in the *master.passwd* file, where the user accounts are traditionally stored. The login classes are listed alongside the group information. The login classes themselves are specified in the *login.conf* file. In the examples, we will list only the MAC-related configuration of the login classes. The MAC labels used in FreeBSD have the syntax of *policy1/qualifier1,policy2/qualifier2,...*, where the policy component describes the policy module (e.g., Biba or MLS), and the qualifier describes a corresponding grade, compartments, and a combined grade-compartment range [8].

Table I  
FOLDER LAYOUT

AccountingGoals	StrategicTechnologyGoals
AccountingReports	SummarySalesReports
SummaryTechnicalReports	TechnicalReports
FeatureRequests	Temp
SalesGoals	UAccountingReports
SalesReports	USalesReports
StrategicSalesGoals	UTechnicalReports

In the following sections the increasingly complex information flow control goals are considered. The goals are then translated into a MAC policy sufficient to fulfill them, and then into a FreeBSD implementation.

### III. PROTECTING THE INTEGRITY

Suppose that the organization needs to protect itself from making decisions based on the untrusted data. That is, John should not make decisions based on the data directly received from Alice or Robert. This corresponds to the John's subject (which is his user login) having a higher integrity label: he should be prevented from inadvertently reading a document of the lower integrity label than his own. Also, John's requests should propagate through the organization unchanged: his documents cannot be tampered with by the subjects with a lower integrity. In the information flow sense, the information is allowed to flow from the higher integrity to the lower integrity objects, but the reverse flow is disallowed. This is the Biba model with the liberal  $\star$ -property [6]. It is directly supported in FreeBSD by the *mac\_biba* module. The assignment of the labels to the files in the case of the Biba model support will look like shown in Table II, and the user label assignment is shown in Table V. The *Temp* folder is excluded from the policy by setting its grade to *equal*, because it serves as a document exchange location.

While the Biba policy enforces the integrity, it also disallows the reverse flow of information. It is obvious that the management should be aware of what is going on in the managed departments, so the reverse flow is required for the considered example. This is achievable in FreeBSD by adding *ranged labels*. The ranged labels allow the subjects to change both their grade and compartment within some range: the grade in the numeric one and the compartment in the set one. The

Table II  
FILES: BIBA

AccountingGoals	biba/2
AccountingReports	biba/5
SummaryTechnicalReports	biba/10
FeatureRequests	biba/2
SalesGoals	biba/2
SalesReports	biba/5
StrategicSalesGoals	biba/5
StrategicTechnologyGoals	biba/5
SummarySalesReports	biba/10
TechnicalReports	biba/5
Temp	biba/equal
UAccountingReports	biba/2
USalesReports	biba/2
UTechnicalReports	biba/2

Table III  
FILES: BIBA AND MLS

AccountingGoals	biba/2,mls/low
AccountingReports	biba/5,mls/low
SummaryTechnicalReports	biba/10,mls/50
FeatureRequests	biba/2,mls/50
SalesGoals	biba/2,mls/low
SalesReports	biba/5,mls/low
StrategicSalesGoals	biba/5,mls/low
StrategicTechnologyGoals	biba/5,mls/50
SummarySalesReports	biba/10,mls/low
TechnicalReports	biba/5,mls/50
Temp	biba/equal,mls/equal
UAccountingReports	biba/2,mls/low
USalesReports	biba/2,mls/low
UTechnicalReports	biba/2,mls/50

Table IV  
FILES: BIBA, MLS AND COMPARTMENTS

AccountingPlans	biba/2,mls/50:1
AccountingReports	biba/5,mls/50:1
CommonTechnicalReports	biba/10,mls/50:2
FeatureRequests	biba/2,mls/50:2
FinancialReports	biba/10,mls/50:1
SalesPlans	biba/2,mls/50:1
SalesReports	biba/5,mls/50:1
StrategicMarketingGoals	biba/5,mls/50:1
StrategicTechnologyGoals	biba/5,mls/50:2
TechnicalReports	biba/5,mls/50:2
Temp	biba/equal,mls/equal
UAccountingReports	biba/2,mls/50:1
USalesReports	biba/2,mls/50:1
UTechnicalReports	biba/2,mls/50:2

Table V  
USERS: BIBA

John	biba/10(10-10)
Jane	biba/5(2-10)
Paul	biba/5(2-10)
Alice	biba/2(2-2)
Mary	biba/2(2-2)
Robert	biba/2(2-2)

Table VI  
USERS: BIBA AND MLS

John	biba/10(10-10),mls/100(100-100)
John.Sales	biba/10(10-10),mls/low(low-low)
John.Engineering	biba/10(10-10),mls/50(50-50)
Jane	biba/5(2-10),mls/low(low-low)
Paul	biba/5(2-10),mls/50(50-50)
Alice	biba/2(2-2),mls/low(low-low)
Mary	biba/2(2-2),mls/low(low-low)
Robert	biba/2(2-2),mls/50(50-50)

Table VII  
USERS: BIBA, MLS AND COMPARTMENTS

John	biba/10(10-10),mls/100:1+2(100-100:1+2)
John.Lower	biba/10(10-10),mls/50:1+2(50-50:1+2)
Jane	biba/5(2-10),mls/50:1(50:1-50:1)
Paul	biba/5(2-10),mls/50:2(50:2-50:2)
Alice	biba/2(2-2),mls/50:1(50:1-50:1)
Mary	biba/2(2-2),mls/50:1(50:1-50:1)
Robert	biba/2(2-2),mls/50:2(50:2-50:2)

syntax of these labels is *biba/effectivegrade:effectivecompartments(lograde:locompartments-higrade:hicompartments)*. If John's subject were assigned the *biba/10(10-10)* label, there would be no way for him to access the document labeled *biba/2*. But if the subject's label were *biba/10(2-10)*, he would be able to downgrade it to the necessary grade of 2 to read the document. For the case of the considered example, this situation is avoided. Instead, we use the notion of a *trusted entity*. This entity is a subject that can promote the document integrity grade, and is trusted by the higher integrity subject. In our example, this subject's user should be able to verify the documents submitted by the subjects with the lower integrity, and if possible and needed, to promote the document integrity grade. Jane and Paul are assigned these privilege and responsibility, and are assigned the label of *biba/5(2-10)*. Thus, Jane can downgrade her integrity grade to 2 to read Mary's report, decide that it can be trusted and is worth for John to see, and then copy it to John while promoting its integrity grade to 10. This is achievable by the following sequence of actions:

- Mary creates a *Report1* file in the *UAccountingReports* folder.
- She moves the file to the *Temp* folder.
- Jane's current integrity label is *biba/5(2-10)*. She downgrades it to *biba/2* using a command like *setpmac biba/2 sh*, and copies the *Report1* file to her home folder. The ownership of the document's copy belongs to Jane.
- Now Jane can read the report and check it for mistakes or forged data.
- After she decides that the document can be forwarded to John, she changes its integrity label to *biba/10* by using the *setfmac biba/10 Report1* command. She then promotes her own clearance to *biba/10* as well, and moves the document to the *SummarySalesReports* folder. That folder has the integrity label of *biba/10* and is accessible to John.
- Now John can read the document.

The described policy protects the integrity of the data, and makes sure that the top management decisions are not affected by any unverified data. We proceed by adding the confidentiality to the organization's information security goals.

#### IV. ADDING CONFIDENTIALITY LEVELS

As the company we consider is a technology one, it is rather possible that it generates the intellectual property (IP). The Engineering department (in our case, Paul and Robert) generates the company's IP. John should have access to it, while Jane and her team should not. However, John may have access to the more confidential information than Paul. For example, he may be working with the company's partners on the subject of a common patent or elaborating a shared strategic technology plan. Also, not only should the confidential documents be kept away from the persons with the lower clearance levels, but the persons with the higher clearance levels should be prevented from disclosing the confidential information they know, either knowingly or inadvertently. The policy to govern this kind of information flow may look like the following:

- A confidential object with the higher security classification cannot be read by a subject with the lower clearance.
- A subject with the higher clearance cannot write to an object with the lower security classification.

This policy matches the one provided by the Bell-LaPadula model with the liberal  $\star$ -property [9] (or the closely related BLP model [6]). The Bell-LaPadula model assigns a clearance label per every confidentiality grade in the organization. It is very similar in its structure to the Biba model. For the considered case, there should be three clearance labels: Secret, Confidential, and Public, with the latter one assigned to Jane and her team. The Bell-LaPadula model is supported by the FreeBSD *mac\_mls* module (further referred to as MLS).

Also, it is still required to preserve the integrity, and do not loose the work done in the previous section. FreeBSD allows this by providing the capability of running multiple MAC policy modules simultaneously. Thus, it is possible for the Biba and the Bell-LaPadula policies to be in effect at the same time, and to preserve both integrity and confidentiality.

For the FreeBSD implementation, we map the Secret clearance to the grade of 100, Confidential to 50, and Public to *low*. The updated file labels are shown in Table III, and the user labels are shown in Table VI.

Compare the user labels assignment for the Biba and the Bell-LaPadula policies. There are two new user logins for the Bell-LaPadula model: *John.Sales* and *John.Engineering*. As per the confidentiality configuration, while being in the CEO position, John cannot directly publish the information for the Sales and the Engineering, for his clearance level is the highest one. By logging in as the lower clearance subject, John will be unable to access his documents, and this will prevent him from inadvertently (or any malicious software on his computer from knowingly) disclosing secret and confidential information to the lower clearance subjects.

## V. SPLITTING THE COMPETENCE AREAS

Jane has the aggregate data of the sales activity of the company. While the data of every individual salesperson cannot give the whole picture, the aggregate reveals the current state of the company sales. Thus, it might be desirable to classify the aggregate information as Confidential. In the considered example, simply adding an MLS grade would not help, because the MLS grades are totally ordered, and the new labels will conflict with the Engineering department: either the engineers will be able to read the Sales' documents, or vice versa (this depends on whose grade is higher). In other words, the Sales and the Engineering should have their own hierarchies with a single point of intersection: John should have access to the documents of both departments. This corresponds to the set-based domination relation: for two sets  $S_1$  and  $S_2$ ,  $S_1$  dominates  $S_2$  (further written as  $S_1 \succ S_2$ ) iff  $S_2 \subset S_1$ . Obviously, incomparable labels are supported by this definition:  $\{1, 2, 3\} \subset \{1, 2, 3, 5\}$ , but  $\{5, 9\}$  and  $\{5, 6, 7\}$  are incomparable. In the considered example, three labels are sufficient:  $S_{\text{Sales}} = \{1\}$ ,  $S_{\text{Eng}} = \{2\}$ , and  $S = \{1, 2\}$ , where label  $S$  belongs to John, so that he can access documents belonging to any of the departments:  $S \succ S_{\text{Sales}}$  and  $S \succ S_{\text{Eng}}$ .

FreeBSD supports the set-based domination relation by the notion of *compartments*. Compartments are the sets of integers, which can be added to the Biba or the MLS labels. In the example, the labels are directly mapped to the compartments. The MLS parts of the labels may be rearranged, as they are not used for distinguishing the Engineering and the Sales any longer. Instead, the MLS grades set the confidentiality grades within individual compartments. The resulting folder and user labels are shown in Tables IV and VII, respectively. The *John.Sales* and *John.Engineering* logins are removed, while a new one is added: *John.Lower*. This new login has the same MLS grade as Jane's and Paul's ones, and is used for creating the documents with their confidentiality classification.

With these settings, John is the only person who accesses the secret information. All other users have access to the confidential information, and none of them deals with creating publicly accessible documents. John has access to both compartments. While it is possible for him to disclose the information of one compartment to another one, this will not violate the confidentiality grades, and John is considered a trusted entity for managing the inter-department information flows.

## VI. THE CHINESE WALL POLICY

The Chinese Wall policy first formalized by Brewer and Nash [10] is oriented toward the commercial sector. This policy provides for prevention of information flows, which cause conflicts of interest for individual consultants working for the same consulting company. As soon as a consultant has obtained access to the confidential information of a client company, he or she must have their access rights to all other companies from this industry sector revoked. However, the consultant must still have access to the public information of all client companies. The consultant starts with the access rights to all information of all client companies. As soon as the consultant accesses the information of a company  $C$  from an industry  $I$ , he is then denied access to the confidential information of all other companies from the industry  $I$ . However, the consultant's access to any information of any company from other industries is allowed. Thus, the conflict of interest does not take place. The deficiencies and important enhancements of the original Chinese Wall policy model are described by Sandhu in [11]. In the same paper, the author demonstrates that the enhanced Chinese Wall policy can be represented in the Bell-LaPadula model, and is therefore a lattice-based policy. While discussing a possible implementation of this policy on FreeBSD, we will follow the definitions given in [6].

Let the classes of conflict of interest (i.e. industries) be represented as a set  $I = \{I_1, I_2, \dots, I_N\}$ , and the number of companies in a single industry assumed equal and be denoted by  $C$ . A clearance label will be then represented by an  $N$ -element vector  $l = \{l[1], l[2], \dots, l[N]\}$ ,  $\forall k \in \overline{1, N} : l[k] \in I_k \cup \{\perp\}$ , where  $\perp$  in position  $k$  denotes the public information of any client company belonging to the industry  $k$  ( $\perp$  is to be read as *null*). For example, a label for an object containing confidential information from the company 3 in the industry 2 and the company 2 from the industry 4 is represented as  $[\perp, 3, \perp, 2, \dots, \perp]$ . There is a dominance relation for these labels defined as follows:  $l^1 \geq l^2$  if  $\forall k \in \overline{1, N} : (l^1[k] = l^2[k]) \vee (l^1[k] \neq \perp \wedge l^2[k] = \perp)$ . Thus,  $[1, 1, 2] > [1, \perp, 2]$  and  $[1, 1, \perp] > [1, \perp, \perp]$ , but  $[1, 1, 2]$  and  $[1, 2, 2]$  are incomparable. The label of all nulls  $[\perp, \dots, \perp]$  is dominated by all other labels. The highest label is denoted by SYSHIGH. The sample lattice for the case of 2 industries and 2 companies per industry is shown in Fig. 2.

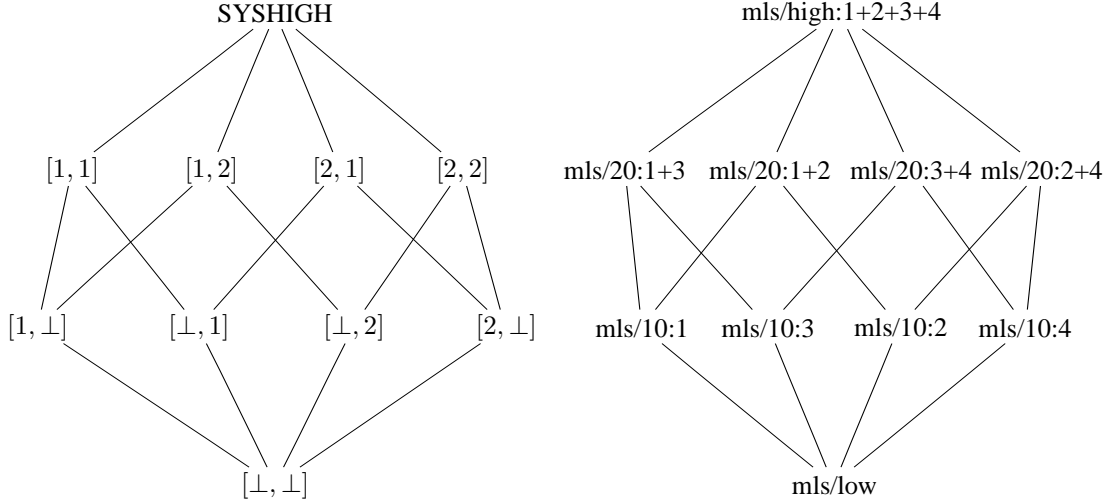


Figure 2. Example Chinese Wall Lattice and Its Mapping to FreeBSD Compartments

It is important to understand how the lattice structure is completed by defining the class-combining join operator  $\oplus$ . Two labels  $l^1$  and  $l^2$  are called *compatible* if  $\forall k \in \overline{1, N} : l^1[k] = l^2[k] \vee l^1[k] = \perp \vee l^2[k] = \perp$ . This also means that all comparable labels are compatible, but there are incomparable labels which are compatible, like  $[1, \perp, 2]$  and  $[\perp, 3, 2]$ . Incompatible labels cannot be combined, while for compatible labels the following rule exists:  $l^1 \oplus l^2 = \{\forall k \in \overline{1, N} : \text{if } l^1[k] \neq \perp \text{ then } l^1[k] \text{ else } l^2[k]\}$ . For example,  $[\perp, 3, 1] \oplus [2, \perp, 1] = [2, 3, 1]$ . We will denote the labels  $l^1$  and  $l^2$  as being compatible by  $\hat{C}(l^1, l^2)$ . Understanding this definition will be required during feasibility analysis for the FreeBSD implementation of the policy.

As a user progresses upwards the domination relation over the lattice, his rights become more and more restrictive. The user leaves a trail of subjects during this progress. For example, for user Mary, the access to public information is granted to the subject “Mary”. As soon as Mary starts working on a project with a bank A, a new subject “Mary.Banks.A” is created for her. When she starts working on a project with an oil company B, another subject is created for her: “Mary.Banks.A.Oil.B”. While in a session with “Mary.Banks.A.Oil.B” subject, Mary can read the public documents from the banking and oil industries, but cannot write to them. To write to the public documents for the oil industry, she has to close her current session and initiate a new one with the subject “Mary”.

The assignment of the user names may be assumed to be performed manually (for example, by a system administrator). The FreeBSD implementation of the Chinese Wall policy is based on this fact. All possible login classes should be created in advance, so that adding a new user will not lead to rebuilding the login classes. The set of login classes required is  $(C + 1)^N$ , and one of those classes is a public one. The number of compartments required is  $C \times N$ . The FreeBSD MAC implementation imposes a limit on the number of compartments: it should be lower than 256. This limitation should be taken into account when checking a specific organization Chinese Wall policy for feasibility. To prevent the user from applying the dominance relation to the classes dominated according to the compartment set inclusion relation, MLS labels are added. The number of MLS grades should be  $N + 1$  (taking into account the public one, and with the exception of the SYSHIGH class).

The rule for labeling the login classes on FreeBSD closely follows the  $\oplus$  operator considered earlier in this section. The labeling algorithm is shown in Fig. 3. As a result of applying this algorithm to the original lattice, a set of triplets is constructed. A triplet contains an MLS grade, a FreeBSD compartments set, and a corresponding original Chinese Wall label. The algorithm iterates over the lattice by examining the set of labels with a fixed non- $\perp$  set of positions on every step. The

previously constructed triplets are examined for containing the original labels dominated by the current one. The corresponding FreeBSD compartment sets are combined and added to the currently constructed triplet.

```

1:  $L^1 \leftarrow \{l : \exists i : (l[i] \neq \perp) \wedge (\forall j \neq i : l[j] = \perp)\}$ ,  $M^1 \leftarrow \emptyset$ ,  $i \leftarrow 1$ 
2: for all  $l \in L^1$  do
3:    $M^1 \leftarrow M^1 \cup \langle 10, i, l \rangle$ 
4:    $i \leftarrow i + 1$ 
5: end for
6: for  $i \leftarrow 2, N$  do
7:    $g \leftarrow \{\text{if } i \neq N \text{ then } i \times 10 \text{ else high}\}$ 
8:    $X^i \leftarrow \text{set of all } i\text{-subsets of } \{1, \dots, N\}$ ,  $L^i \leftarrow \emptyset$ ,  $M^i \leftarrow \emptyset$ 
9:   for all  $x \in X^i$  do  $L^i \leftarrow L^i \cup \{l : (\forall j \in x : l[j] \neq \perp) \wedge (\forall j \notin x : l[j] = \perp), j = \overline{1, N}\}$ 
      Now  $L^i$  contains all lattice node labels with  $i$  positions being not  $\perp$ .
10:  end for
11:  for all  $l \in L^i$  do
12:     $M' \leftarrow \{\forall m' \in M^{i-1} : m' = \langle g', f', l' \rangle \wedge \hat{C}(l, l')\}$ 
13:     $M^i \leftarrow M^i \cup \{\langle g, \bigcup_{m' \in M'} f', l \rangle\}$ 
14:  end for
15: end for
16:  $M \leftarrow \bigcup_i M^i$ 

```

Figure 3. FreeBSD Label Generation for Chinese Wall Policy

The system administrator has to create a set of login classes for the combined Bell-LaPadula and compartment labels. When a user starts working with a company from a new industry, the administrator provides the user name corresponding to the changed Chinese Wall lattice label.

## VII. CONCLUSION

The set of the policies implemented by the FreeBSD/TrustedBSD project allows to implement not only the Bell-LaPadula and Biba models. By reusing the result of the Chinese Wall model being a lattice-based one, it can be implemented in the FreeBSD MAC framework as well. Also, the level of MAC support allows implementing trusted entities, so that the information flows are possible both ways, but in a controllable manner, so that the policies are not violated. The combination of a number of MAC models provides a very flexible platform for satisfying the needs of various organizations. The authors hope that the proper illustration of FreeBSD MAC support capabilities will promote the usage of mandatory access control in the commercial sector.

## ACKNOWLEDGMENT

The authors are grateful to Alexey Fomin whose comments and suggestions have significantly improved the paper.

## REFERENCES

- [1] "FreeBSD project. FreeBSD home page," <http://www.FreeBSD.org/>.
- [2] R. N. M. Watson, "TrustedBSD: Adding trusted operating system features to FreeBSD." in *USENIX Annual Technical Conference, FREENIX Track*, C. Cole, Ed. USENIX, 2001, pp. 15–28.
- [3] R. Watson, W. Morrison, C. Vance, and B. Feldman, "The TrustedBSD MAC framework: Extensible kernel access control for FreeBSD 5.0." in *USENIX Annual Technical Conference, FREENIX Track*. USENIX, 2003, pp. 285–296.
- [4] D. E. Denning, "A lattice model of secure information flow." *Commun. ACM*, vol. 19, no. 5, pp. 236–243, 1976.
- [5] *Department of Defense Trusted Computer System Evaluation Criteria. DoD 5200.28-STD*, Department of Defense National Computer Security Center, 1985.
- [6] R. S. Sandhu, "Lattice-based access control models." *IEEE Computer*, vol. 26, no. 11, pp. 9–19, 1993.
- [7] S. L. Osborn, "Information flow analysis of an RBAC system." in *SACMAT*, 2002, pp. 163–168.
- [8] *The FreeBSD Documentation Project. FreeBSD Handbook*, The FreeBSD Project, [http://www.FreeBSD.org/doc/en\\_US.ISO8859-1/books/handbook/](http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/handbook/).
- [9] D. E. Bell and L. J. LaPadula, "Secure computer systems: A mathematical model, volume II." *Journal of Computer Security*, vol. 4, no. 2/3, pp. 229–263, 1996.
- [10] D. F. C. Brewer and M. J. Nash, "The Chinese Wall security policy." in *IEEE Symposium on Security and Privacy*, 1989, pp. 206–214.
- [11] R. S. Sandhu, "Lattice-based enforcement of Chinese Walls." *Computers & Security*, vol. 11, no. 8, pp. 753–763, 1992.