ScienceDirect

# Lapadula Model

Related terms:

Media Access Control, Access Control Policies, Security Model, Classification, Biba Model, Discretionary Access Control, Mandatory Access Control

## Policies, Access Control, and Formal Methods

Elisa Bertino, in Handbook on Securing Cyber-Physical Critical Infrastructure, 2012

### 23.2.2 Mandatory Access Control Model

Unlike the access control models based on the notion of access control matrix, in the mandatory access control, the access control decisions are based on specific relationships between the subject requesting access and the object to which access is requested. An important motivation for the mandatory access control is to control the flow of information once the information has been accessed. Access control mechanisms based on the notion of access control matrix typically only control whether each single access is authorized; however, they do not control where the data flow once have been accessed. These access control mechanisms are thus unable to protect against "Trojan Horses." A "Trojan Horse" is a piece of code embedded into an application program. When the program is running on behalf of a subject, the "Trojan Horse" exploits the authorizations of this subject in order to gain access to protected data and transfer the data into some other objects accessible to subjects not authorized to access the protected data. Access control mechanisms based on the mandatory access control prevent such attacks.

The most well-known mandatory access control model was proposed by Bell and LaPadula [8]; later Denning [9] generalized the notion of mandatory access control into the notion of lattice-based access control, also known as information flow control. Under a mandatory access control model, the action of accessing a data object starts an information flow. In particular, reading a data object causes information to flow from the data object to the subject, whereas the flow is in the opposite direction if the subject writes to the data object. A mandatory access control model thus consists of rules specifying which information flows are authorized.

In the Bell–LaPadula (BLP) model the information flows are authorized based on comparing a certain specific property of subjects and data objects. This property, referred to as *access class*, is associated with each subject and data object. The access class of a data object indicates the sensitivity of the data object, whereas the access class of the subject indicates how much the subject can be trusted not to disclose sensitive information.

In the BLP model, each access class consists of two elements, and the set of all access classes is partially ordered according to a relation called *dominance relation*, denoted by ≥. Accesses to data objects by subjects are regulated by two properties:

1. The simple security property (*no-read-up*): A subject can read a data object if its access class dominates the access class of the data object.
2. The *-property (*no-write-down*): A subject can write into a data object if its access class is dominated by the access class of the data object.

For a system to be secure both properties must be verified by any system state.

In many practical situations, however, the two properties, especially the *-property, are too restrictive. For example, a trusted user may be allowed to access some sensitive data and, perhaps after sanitizing it, to transfer it into some unclassified data object. To address this problem the model provides a mechanism by which each subject has a *maximum access class* and a *current access class*. A subject may change its access class; however, its current access class must at any time be dominated by the maximum access class.

This model has been implemented in several systems, including operating systems and DBMSs. A notable example is represented by Oracle Label Security, a relational DBMS product in which access classes, called *labels*, consist of three components and which also provides a comprehensive environment for the management of labels and associations of labels with data objects and users.

Read full chapter

URL: https://www.sciencedirect.com/science/article/pii/B9780124158153000236

## Security for Distributed Systems: Foundations of Access Control

Elisa Bertino, Jason Crampton, in Information Assurance, 2008

### 3.3.4 Bell-LaPadula Model

Clearly, it is unlikely that an information flow policy will be able to define all the authorization requirements of a computer system. Notice that an information flow policy authorizes access purely on the basis of labeling of the subject and object, and is independent of any ownership considerations. This is sometimes referred to as a *mandatory* access control policy. In practically all cases, no user should actually be authorized to write to a file with a higher security label, but the information flow policy does not prevent this from happening. Therefore, it is necessary to augment the mandatory information flow policy with a *discretionary* access control policy, defined by the object owners and implemented using a protection matrix.

The Bell-LaPadula model is perhaps the most famous of all access control models and has had a significant influence on the development of research into access control. It allows for the definition of a mandatory information flow policy and a discretionary access control policy. The first contribution of the Bell-LaPadula model was to formally define what it meant for a computer system to be in a *secure state*. A second contribution was to prove that it is possible to construct computer systems that only exist in secure states. That is, it is possible to build a computer system and define a security policy such that for all future points in time, the system is in a secure state.

The Bell-LaPadula model implements an information policy for confidentiality, and includes a protection matrix to further refine the information flow policy. The *protection state*, or simply *state*, of a computer system is a snapshot of all security-relevant information that is subject to change. The Bell-LaPadula model defines the state of the system to be $(V, M)$, where $V$ denotes the set of *active triples*.[8] $V$ models those requests that have been granted and represents the set of objects

that are in main memory currently in use by subjects. For example, if a request from *s* to read *o* is granted then the object is brought into main memory, which is modeled by adding the triple (*s, o*, read). It will become apparent that this is an important consideration in checking access requests.

The Bell-LaPadula model defines three security properties. All triples in *V* must satisfy the simple security property and *-property. In addition, every triple in *V* must satisfy the *discretionary security property*, which requires that every granted request has been authorized by the protection matrix.

A state is said to be *secure* if it satisfies the discretionary, simple, and *-properties. In order to maintain the system in a secure state, it must not be possible to grant a request that would violate any of the three security properties. Hence, when a user makes an access request, the system must check both the access matrix to see that the request is authorized by the matrix and compare the security labels of the subject and object to determine whether the simple security property and *-property are satisfied.

The operating system is responsible for all state transitions and must ensure that each command that causes a state transition moves the system from a secure state to another secure state. Therefore, the operating system will contain a number of functions that implement security checks to ensure that the state transition results in a state that conforms to the security policy of the system.

Read full chapter

URL: https://www.sciencedirect.com/science/article/pii/B9780123735669500057

# Information Gathering

Craig Wright, in The IT Regulatory and Standards Compliance Handbook, 2008

## Restrictions with the Bell-LaPadula Model

The Bell-LaPadula Model imposes the following restrictions to object access by subjects:

> **Reading down:** A subject has only read access to objects whose security level is below the subject's current clearance level. This was designed to prevent subjects from accessing information available to higher security clearance levels than the subject has been currently assigned.
>
> **Writing up:** A subject is granted append access to those objects with a security level that is set to be higher than its current clearance level. This was designed to prevent subjects from transitioning information across into those levels that are set with a lower security than the subject's current level.

The Bell-LaPadula model enhances an access matrix with the restrictions listed above in order to afford access control and information flow capabilities. In the event that a subject has been assigned read access to an object in the access matrix, it may be restricted from exercising this right if the object is designated to a security level higher than the clearance level assigned to the subject.

Read full chapter

URL: https://www.sciencedirect.com/science/article/pii/B9781597492669000059

# Domain 3

Eric Conrad, ... Joshua Feldman, in Eleventh Hour CISSP® (Third Edition), 2017

### Bell-LaPadula Model

The *Bell-LaPadula* model was originally developed for the US Department of defense (DoD). It is focused on maintaining the confidentiality of objects. Protecting confidentiality means users at a lower security level are denied access to objects at a higher security level.

### Fast Facts

Bell-LaPadula includes the following rules and properties:

- *Simple Security Property*: "No read up"; a subject at a specific clearance level cannot read an object at a higher classification level. Subjects with a Secret clearance cannot access Top Secret objects, for example.
- *Security Property*: "No write down"; a subject at a higher clearance level cannot write to a lower classification level. For example: subjects who are logged into a Top Secret system cannot send emails to a Secret system.
- *Strong Tranquility Property*: Security labels will not change while the system is operating.
- *Weak Tranquility Property*: Security labels will not change in a way that conflicts with defined security properties.



## Authorization and Access Control

Jason Andress, in The Basics of Information Security, 2011

### Multilevel Access Control

Multilevel access control models are used where the simpler access control models that we just discussed are considered to not be robust enough to protect the information to which we are controlling access. Such access controls are used extensively by military and government organizations, or those that often handle data of a very sensitive nature. We might see multilevel security models used to protect a variety of data, from nuclear secrets to protected health information (PHI).

The Bell-LaPadula model implements a combination of DAC and MAC access controls, and is primarily concerned with the confidentiality of the resource in question. Generally, in cases where we see DAC and MAC implemented together, MAC takes precedence over DAC, and DAC works within the accesses allowed by the MAC permissions. For example, we might have a resource that is classified as secret and a user that has a secret level of clearance, normally allowing them to access the resource under the accesses allowed by MAC. However, we might also have an additional layer of DAC under the MAC access, and if the resource owner has not given the user access, they would not be able to access it, despite the MAC permissions. In Bell-LaPadula, we have two security properties that define how information can flow to and from the resource [5]:

- *The Simple Security Property*: The level of access granted to an individual must be at least as high as the classification of the resource in order for the individual to be able to access it.
-

*The *Property*: Anyone accessing a resource can only write its contents to one classified at the same level or higher.

These properties are generally summarized as "no read up" and "no write down," respectively. In short, this means that when we are handling classified information, we cannot read any higher than our clearance level, and we cannot write classified data down to any lower level.

The Biba model of access control is primarily concerned with protecting the integrity of data, even at the expense of confidentiality. Biba has two security rules that are the exact reverse of those we discussed in the Bell-LaPadula model [6]:

- *The Simple Integrity Axiom:* The level of access granted to an individual must be no lower than the classification of the resource.
- *The *Integrity Axiom:* Anyone accessing a resource can only write its contents to one classified at the same level or lower.

We can summarize these rules as "no read down" and "no write up," respectively. This may seem completely counterintuitive when we consider protecting information, but remember that we have changed the focus from confidentiality to integrity. In this case, we are protecting integrity by ensuring that our resource can only be written to by those with a high level of access and that those with a high level of access do not access a resource with a lower classification.

The Brewer and Nash model, also known as the Chinese Wall model, is an access control model designed to prevent conflicts of interest. Brewer and Nash is commonly used in industries that handle sensitive data, such as that found in the financial, medical, or legal industry. Three main resource classes are considered in this model [7]:

- *Objects*: Resources such as files or information, pertaining to a single organization.
- *Company groups*: All objects pertaining to a particular organization.
- *Conflict classes*: All groups of objects that concern competing parties.

If we look at the example of a commercial law firm working for companies in a certain industry, we might have files that pertain to various individuals and companies working in that industry. As an individual lawyer at the firm accesses data and works for different clients, he could potentially access confidential data that would generate a conflict of interest in him while working on a new case. In the Brewer and Nash model, the resources and case materials that the lawyer was allowed access to would dynamically change based on the materials he had previously accessed.

Read full chapter

URL: https://www.sciencedirect.com/science/article/pii/B9781597496537000037

# Authorization and Access Control

Jason Andress, in The Basics of Information Security (Second Edition), 2014

## Multilevel access control

Multilevel access control models are used where the simpler access control models that we just discussed are considered to not be robust enough to protect the information to which we are controlling access. Such access controls are used extensively by military and government organizations, or those that often handle data of a very sensitive nature. We might see multilevel security models used to

protect a variety of data, from nuclear secrets to protected health information (PHI).

The Bell–LaPadula model implements a combination of DAC and MAC and is primarily concerned with the confidentiality of the resource in question. Generally, in cases where we see DAC and MAC implemented together, MAC takes precedence over DAC, and DAC works within the accesses allowed by the MAC permissions. For example, we might have a resource that is classified as secret and a user that has a secret level of clearance, normally allowing them to access the resource under the accesses allowed by MAC. However, we might also have an additional layer of DAC under the MAC access, and if the resource owner has not given the user access, they would not be able to access it, despite the MAC permissions. In Bell–LaPadula, we have two security properties that define how information can flow to and from the resource [5]:

1. *The simple security property*: The level of access granted to an individual must be at least as high as the classification of the resource in order for the individual to be able to access it.
2. *The * property*: Anyone accessing a resource can only write its contents to one classified at the same level or higher.

These properties are generally summarized as "no read up" and "no write down," respectively. In short, this means that when we are handling classified information, we cannot read any higher than our clearance level, and we cannot write classified data down to any lower level.

The Biba model of access control is primarily concerned with protecting the integrity of data, even at the expense of confidentiality. Biba has two security rules that are the exact reverse of those we discussed in the Bell–LaPadula model [6]:

- *The simple integrity axiom*: The level of access granted to an individual must be no lower than the classification of the resource.
- *The * integrity axiom*: Anyone accessing a resource can only write its contents to one classified at the same level or lower.

We can summarize these rules as "no read down" and "no write up," respectively. This may seem completely counterintuitive when we consider protecting information, but remember that we have changed the focus from confidentiality to integrity. In this case, we are protecting integrity by ensuring that our resource can only be written to by those with a high level of access and that those with a high level of access do not access a resource with a lower classification.

The Brewer and Nash model, also known as the Chinese Wall model, is an access control model designed to prevent conflicts of interest. Brewer and Nash is commonly used in industries that handle sensitive data, such as that found in the financial, medical, or legal industry. Three main resource classes are considered in this model [7]:

1. *Objects*: Resources such as files or information, pertaining to a single organization.
2. *Company groups*: All objects pertaining to a particular organization.
3. *Conflict classes*: All groups of objects that concern competing parties.

If we look at the example of a commercial law firm working for companies in a certain industry, we might have files that pertain to various individuals and companies working in that industry. As an individual lawyer at the firm accesses data and works for different clients, he could potentially access confidential data that would generate a conflict of interest while working on a new case. In the Brewer and Nash model, the resources and case materials that the lawyer was

allowed access to would dynamically change based on the materials he had previously accessed.

Read full chapter

URL: https://www.sciencedirect.com/science/article/pii/B9780128007440000038

## Frustration Strategies

Timothy J. Shimeall, Jonathan M. Spring, in Introduction to Information Security, 2014

### Confidentiality Models

Confidentiality, as described in Chapter 1, is one of the core properties on which security is based. If an organization cannot prevent unauthorized disclosure of information, then it is difficult for that organization to retain control over the use of that information. When the information is critical enough, a clear and unambiguous structure for the analysis of confidentiality becomes useful. This section describes two such structures: Bell–LaPadula, and Chinese Wall.

The Bell–LaPadula model [2] has both mandatory and discretionary components for expressing confidentiality properties in computer systems. Each object in the system has a label expressing its degree of confidentiality, and this label may not be changed or removed from the object (the "tranquility principle"). Each subject has both a clearance level and a current confidentiality level, which may not exceed the clearance level, and is no lower than the maximum confidentiality of the information that has been read. Bell and LaPadula express algebraic semantics in a state-machine form, then define a number of security axioms. Figure 6.5 illustrates these axioms. For mandatory access control, the two most important axioms are the:

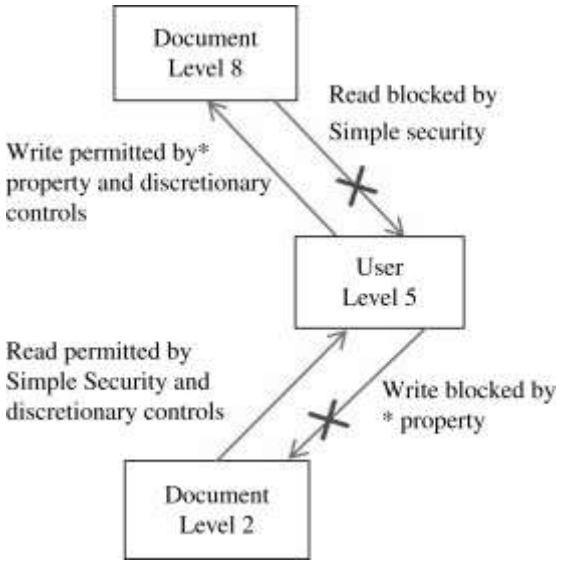


Figure 6.5. Bell–LaPadula properties.

- *Simple security property:* No subject has read access to an object with a classification level higher than the clearance level of the subject.
- **-property ("star property"):* No subject may write to an object with a classification level lower than the current confidentiality level of the subject.

The first property prevents an actor from reading information at a level the subject isn't cleared for (or, colloquially, "no read up"). The second property prevents an actor from declassifying information for unauthorized dissemination (or, colloquially, "no write down"). The discretionary portion of the Bell–LaPadula model uses access control matrices similar to the Graham–Denning model, but with increased stringency to support the two central properties. The Bell–LaPadula model was successfully applied to several secure developments, including secure Xenix [8]. It forms the basis for many of the criteria used for certifying confidentiality, including the common criteria.

The Chinese Wall security model [14] is a formal logic model that takes a different approach to confidentiality than Bell–LaPadula. In the Chinese Wall model, the set of objects on a computer system is partitioned into conflict classes, where a conflict class is defined to be objects that relate to information from competing sources. For example, if an organization is receiving bids from multiple vendors, then the information on a given vendor is bound into a company subset, and the collection of company subsets for a given contract form a conflict class. If a subject may access multiple company subsets for the same contract, then the potential for information leakage (violating the competition) may occur. The model allows for sanitization of information, which involves transformation such that the source of the information may no longer be deduced from the information.

The Chinese Wall security model provides properties preventing such violations:

- Once a subject has accessed any object in a conflict class, then the subject may only access objects that belong to the same company subset, or that belong to an entirely separate conflict class. This implies that a subject may at most have access to one company subset in each conflict class and that if, for some conflict class *X*, there are at least *Y* company subsets, then at least *Y* subjects are required to process *X*.
- A subject may write to an object if the subject has accessed only information in compliance with the preceding property, and if no object containing unsanitized information has been read in another company subset in the same conflict class. This implies that unsanitized information remains contained in the company subset, but sanitized information may be processed freely.

Figure 6.6 visualizes these properties. The Chinese Wall security model has been incorporated in a number of audit models and applied to commercial data processing systems. Figure 6.7 contrasts the Bell–LaPadula and Chinese Wall confidentiality models.

Figure 6.6. Chinese Wall security model properties.

Figure 6.7. Bell-LaPadula contrasted to Chinese Wall.

## Profile

### Roger Schell

Roger Schell was born in the 1930s in eastern Montana [15]. He grew interested in electronics working on radios at an early age, which grew into an academic career in electrical engineering with graduate degrees at Washington State and MIT. In the course of his graduate studies he was more and more involved with computers, eventually working on the development of MULTICS while at MIT. Exposure to some prominent formal methods professors lead him to experiment and master formal derivation approaches, although formal security models did not exist at the time.

After his Ph.D., he returned to the U.S. Air Force and eventually became interested in computer security. He developed the concept of the security kernel, and used that as a basis for making formal statements about the security of an operating system. This design approach, in turn, lead him to experience in developing security models and evaluating operating systems. This experience eventually allowed him to spearhead the development of the influential "Trusted Computer System Evaluation Criteria," colloquially known as the "Orange Book" for the brightly colored covers given it in production.

After leaving the U.S. Air Force, Schell became active in industry, leading companies that build trustable computing systems. He currently remains active in such efforts.

There are a number of other confidentiality models that have been developed (see, for example, the military need-to-know model described in Chapter 9). However, these should provide an insight into how such models constrain the flow of information in a computer system to provide verifiable confidentiality.

Read full chapter

URL: https://www.sciencedirect.com/science/article/pii/B9781597499699000067

# Domain 5

Eric Conrad, in Eleventh Hour CISSP, 2011

## Security models

Now that we understand the logical, hardware, and software components required to have secure systems, and the risk posed to them by vulnerabilities and threats, we can move on to security models, which provide rules for secure system operation.

### Bell-LaPadula model

The **Bell-LaPadula** model was originally developed for the Department of Defense. It is focused on maintaining the confidentiality of objects. Protecting confidentiality means *not* allowing users at a lower security level to access objects at a higher security level.

### Lattice-based access controls

**Lattice-based access control** allows security controls for complex environments. For every relationship between a subject and an object, there are defined upper and lower access limits implemented by the system. This lattice, which allows reaching higher and lower data classification, depends on the need of the subject, the label of the object, and the role the subject has been assigned. Subjects have a Least Upper Bound (LUB) and Greatest Lower Bound (GLB) of access to the objects based on their lattice position. Figure 5.3 shows a lattice-based access control model. At the highest level of access is the box labeled {Alpha, Beta, Gamma}. A subject at this level has access to all objects in the lattice.

Figure 5.3. Lattice-based access control.

Fast Facts

Bell-LaPadula includes the following rules and properties:

- *Simple Security Property*. "No read up": a subject at a specific classification level cannot read an object at a higher classification level. Subjects with a "Secret" clearance cannot access "Top Secret" objects, for example.

- *Security Property*. "No write down": a subject at a higher classification level cannot write to a lower classification level. For example, subjects who are logged into a Top Secret system cannot send emails to a secret system.
- *Strong Tranquility Property*. Security labels do not change while the system is operating
- *Weak Tranquility Property*. Security labels do not change in a way that conflicts with defined security properties

At the second tier of the lattice, we see that each object has a distinct upper and lower allowable limit. For example, if a subject has {Alpha, Gamma} access, the only viewable objects in the lattice are the Alpha and Gamma objects. Both represent the greatest lower boundary. The subject would not be able to view object Beta.

### Integrity models

Models such as Bell-LaPadula focus on confidentiality, sometimes at the expense of integrity. The Bell-LaPadula "No Write Down" rule means that subjects can write up: A secret subject can write to a top secret object. What if the secret subject writes erroneous information to a top secret object? Integrity models such as Biba address this issue.

## Biba model

While many governments are primarily concerned with confidentiality, most businesses want to ensure that the integrity of the information is protected at the highest level. **Biba** is the model of choice when integrity protection is vital.

Fast Facts

The Biba model has two primary rules: the Simple Integrity Axiom and the *Integrity Axiom.

- *Simple Integrity Axiom*. "No read down": A subject at a specific classification level cannot *read* data at a lower classification. This prohibits subjects from accessing information at a lower integrity level, thus protecting integrity by preventing bad information from moving up from lower integrity levels.
- **Integrity Axiom*. "No write up": a subject at a specific classification level cannot *write* to data at a higher classification. This prevents subjects from passing information up to an integrity level higher than the one they have clearance to change. In this way integrity is protected by preventing bad information from moving up to higher integrity levels.

Biba is often used where integrity is more important than confidentiality. Examples include time- and location-based information.

Did you Know?

Biba takes the Bell-LaPadula rules and reverses them, showing how confidentiality and integrity are often at odds. If you understand Bell-LaPadula (no read up; no write down), you can extrapolate Biba by reversing the rules (no read down; no write up).

## Clark-Wilson

Clark-Wilson is a real-world model that protects integrity by requiring subjects to access objects via programs. Because the programs have specific limits on what

they can and cannot do to objects, this model effectively limits the capabilities of the subject.

Clark-Wilson uses well-formed transactions to provide integrity. This concept comprises the "access control triple": user, transformation procedure, and constrained data item.

### Chinese Wall model

The Chinese Wall model is designed to avoid conflicts of interest by prohibiting one person, such as a consultant, from accessing multiple conflict-of-interest (CoI) categories. It is also called Brewer-Nash, after its creators, Dr. David Brewer and Dr. Michael Nash, and was initially designed to address the risks inherent in employing consultants in banking and financial institutions.[1]

### Access control matrix

An access control matrix is a table defining the access permissions that exist between specific subjects and objects. A matrix is a data structure that acts as a table lookup for the operating system. For example, Table 5.1 is a matrix that has specific access permissions defined by users and detailing what actions they can enact. User BLakey has read/write access to the data file as well as access to the data creation application. User AGarner can read the data file and still has access to the application. User CKnabe has no access within this data access matrix.

Table 5.1. User Access Permissions

| Users | Data Access File #1 | Data Creation Application |
|---|---|---|
| BLakey | Read/Write | Execute |
| AGarner | Read | Execute |
| CKnabe | None | None |

Read full chapter

URL: https://www.sciencedirect.com/science/article/pii/B9781597495660000059

## Control of Information Distribution and Access

Ralf Hauser, in Advances in Computers, 1997

### 4.1 The Relation between Access Control and Usage Control

Discretionary access control is not aware of what happens after access has been granted, that is, after information has been disclosed. It is therefore stateless. This type of access control was feasible without specific usage control features in the past for the following reasons:

- Termination of usage is assumed to take place when the user eventually quits the application, or in the event of system logout of the user or system reboot.
- Proliferation of binaries was primarily protected by legal contracts. It was also slower than today, because if the user domain was connected to an external network at all, the transmission capacity was small compared to the code size.
- The administrative scope of file systems was very limited (few and restricted "intercampus" file systems).

Mandatory AC (MAC) schemes prevent information from flowing to lower classified system parts even after access has been granted. The Bell–LaPadula model maintains state about security level of the objects currently accessed, but still does not provide features necessary for effective usage control—it doesn't control the following:

1. The number of accesses from the same clearance level
2. The duration of the access
3. Local copies and potential alterations thereof

Therefore, it is not possible to impose any additional cost on the user when an information recipient "uses" some information asset simultaneously twice.

#### 4.1.1 ORCON

Originator controlled release (ORCON) is an extension of the MAC schemes. In the conventional paper world, there exist numerous dissemination/handling restrictions called "release markings" that are in place in the DOD/intelligence community or the Bureau of Labor Statistics. The computerized version of this idea extends the previously described concepts of access control in two ways:

- The task of the security administrator is split into two distinct subtasks: An administrator on the information originator side (ORGREP) and one on the information recipient side (RECREP) [61].
- Not only can the owner specify access restriction when initially releasing some information, but also he or she still may change, for example, an access control list post-release, while the object is floating around in the system in various instances. The access control list remains under his or her full jurisdiction, and all instances of the object, independent of whether they are physically owned by him or her, will obey the new decision. This is called Owner Retained Access Control (ORAC) [62].

The ORGREP is the security administrator or representative of the originator of the coded information. The ORGREP has the authority to change the access control information (ACI) of the concerned information object. The RECREP role is the representative of the users of the code. The RECREP has the authority to change the ACI of the context input to the ADF, that is, subjects' (users') group memberships, their privileges, etc.

This leads to a classification of AC systems in a three-dimensional space. The first dimension is whether the subject (user) has the authority to change the ACI of objects (code) he or she owns, or whether this right is reserved for some ORGREP, such as a license issuer. The second dimension is whether the system controls the ACI of an object when it enters a new subject's realm (propagation of ACI to a new owner, for example, enforcing licensing schemes), and the third dimension is whether an entire domain has to be structured in a uniform way. Hence, Graubart [63] classified MAC, DAC, and ORCON as shown in Table III. Even ORAC/ORGCON are not concerned about the number of simultaneous accesses and their duration. Also, mechanisms to remove code from primary memory at the client's side of a distributed file system (DFS) are normally not in place.

Table III. Classification of Access Control Mechanisms with Respect to Protection Policies

| AC type | Object ACI Manipulation by Subject | Trans-Subject-Realm Control | System Flexibility |
|---|---|---|---|
| DAC | Changeable | No propagation | Tailorable, no uniform, central policy |
| MAC | Not changeable | Propagation | Uniform, central policy |
| ORCON | Not changeable | Propagation | Tailorable extensions to a uniform, central policy |

Read full chapter

URL: https://www.sciencedirect.com/science/article/pii/S006524580860340X

# Security analysis of computer networks

Gürkan Gür, ... Fatih Alagöz, in Modeling and Simulation of Computer Networks and Systems, 2015

## 6.1 Formal security analyses

The formal modeling and verification approach is one of the most effective tools for network security analysis with maximal precision [43]. It serves the common goals of network security analysis, which are to improve the quality of the system specification and to check for the existence of security deficiencies. Moreover, it may enable a more systematic understanding of security issues and facilitate systematic testing of network-related implementations.

A security model is a formal description of security related aspects and mechanisms of a system using formal methodology. A model is formal if it is specified using a formal language, which is defined as a language with well-defined syntax and semantics such as finite state automata (FSM) and predicate logic [43]. That model includes a base system component and a security component related with a satisfaction requirement as shown in Figure 30.9. The former defines what the system does while the security component is an abstraction of security requirements. The satisfaction relation ensures that these security requirements are met and typically verified via formal methods. Therefore, security analysis is carried out based on correspondence between system description and security properties adopted in the formal modeling.

Figure 30.9. The structure of a formal security model [44].

According to [43], there are four classes of practically relevant formal security models:

1. **Automata Models:** A model checking specification consists of two parts [45]. One part is the model: a state machine defined in terms of variables, initial values for the variables, and a description of the conditions under which variables may change value. The second part is temporal logic constraints

defined over states and execution paths. Conceptually, a model checker visits all reachable states and verifies that the temporal logic properties are satisfied over each possible path, that is, the model checker determines if the state machine is a model for the temporal logic formula via exploration of the state space temporal logic constraints over states and execution paths. For the analysis to be performed, the constituent elements of the model such as vulnerability description, connectivity and required function need to be developed. In [46], Mao et al. describe a new approach, namely logical exploitation graphs, to represent and analyze network vulnerability. Their logical exploitation graph generation tool illustrates logical dependencies among exploitation goals and network configuration. Their approach reasons all exploitation paths using bottom-up and top-down evaluation algorithms in the Prolog logic programming engine.

2. Access Control Models: Classical access control models, like the traditional Bell-LaPadula model [43], relying on access control rules with security labels on objects and clearances for users, lack the modeling capabilities for current practical systems. Role-based access control (RBAC) models mapping subjects to roles in a hierarchical structure and then relating roles to access rights to subjects are proposed to address this shortcoming. In [47], a formal model of the computer network is constructed using graph-theoretic tools with packet filter functions classifying the message flow thus constraining the reachability of the entities on the network. Access control lists and routing policies are reflected into the model by means of packet filtering functions that are associated with edges of the graph.
3. **Information Flow Models:** These models describe how information may flow between which domains in a very abstract way such that they can capture also indirect and partial flow of information [48,49]. An example is the confidentiality of data output from a system. The critical issue is not whether any output contains confidential data but rather depends on it [44]. The concept of *noninterference* is a fundamental information flow property in that regard. Basically, if there is no information flow from one group of processes to another, the first group is said to be noninterfering with the other. This means that the processes in the first group cannot reveal any secret information such as passwords or encryption keys to the entities in the second group. In return, the processes in the second group cannot be corrupted by the ones in the first group. This expressive capability provides the basis for modeling confidentiality and integrity requirements between processes.
4. **Cryptoprotocol models:** Probably the most successful class of security models are cryptoprotocol models describing the message traffic of security protocols. The formal and mathematical design of cryptographic schemes is suitable for formal verification. Virtually all formal methods have been employed for cryptoprotocol verification [50] extending to industrial size protocols. Mostly secrecy and authentication goals can be specified and then verified automatically using model-checkers tailored for this application.

Formal methods can only address certain aspects of security, those related to computer networks and system design. Some aspects of security do not lend themselves to formal methods (e.g., computer hacking, tampering, and social engineering) [51]. Since current ICT systems are very complex, it is also nontrivial to have methods scaling with network size and security flaws and vulnerabilities are hard to find using formal analysis. In Figure 30.10, a cost-benefit analysis for formal methods is shown. The benefit is a function of number of users and their importance level, i.e., as the number of users and their relative importance increase, the investment on formal analysis is more reasonable. On the difficulty

aspect, the cost increases with increasing system and property complexity. The operation domain of the system renders formal analysis feasible or infeasible due to complexity. Event it is feasible, it may unreasonable due to small return-on-investment for the analysis efforts. Therefore, security analysis based on formal methods typically focuses on specific aspects of system in operation. However, these analyses are precise and can be automated.

Figure 30.10. The cost-benefit analysis for formal methods [52].