

Latest modified: September 4, 2003

Chapter 3. Policies

The need for policies

Every institution has a set of business policies, explicit or implicit. In addition, having a set of security policies is fundamental. Without this, it is impossible to set up secure systems, we don't know what we should protect and how much effort we should put on security. Effectively, a security policy divides the states of a system into authorized and non-authorized states. That means, that we cannot even talk about security without policies because we don't know what states we should avoid.

The institution security policy includes laws, rules, and practices that regulate how an institution manages and protects resources. Another definition is: high-level guidelines concerning information security [Woo80]. Some of these policies may come from external sources; e.g., legislation, government or industry standards. There are some interesting issues about who should define the external policies [Mea00] or what standards we should adopt.

The computer systems at the institution should enforce these policies by defining specific versions of the security policies. The layers pattern can be used to describe how the policies are structured, going from the institution policies at the highest level down to policies to allow specific accesses of users to data, encryption policies, etc.

Note that some people call policies what we call models. In our approach, which follows classical security work, models implement policies, not define policies. The need and the value of policies have been recognized lately and now there is even an annual conference dedicated to policies [Pol].

Security policies

Through experience several policies have evolved as the most convenient to build secure systems. We enumerate some of these here:

- ?? *Open/closed systems*—In a closed system, nothing is accessible unless explicitly authorized, in an open system or institution everything is accessible unless explicitly denied. Clearly a secure system must be closed. Institutions where information secrecy and integrity are important, e.g., banks, use a closed policy, while institutions such as libraries use an open policy.
- ?? *Least privilege (need to know)*—People/systems should be authorized only for resources they need to perform their functions. This policy is usually combined with the closed system policy.
- ?? *Authorization*—Explicit rules must be used to define who can use what resources and how. Authorizations may allow or deny access.
- ?? *Separation of duty*—Critical functions should be divided between people or systems. For example, the person who decides purchasing of a product is not the same who actually orders the product.

- ?? *Auditing*—An audit trail should be kept recording what was done at what time. This will help to prevent future attacks.
- ?? *Centralized/decentralized control*. In a decentralized system its units or divisions have authority to define their own policies as far as they don't violate global policies.
- ?? *Individual accountability*—People or processes must be uniquely identified and their actions are recorded and reviewed.
- ?? *Roles*. Roles imply sets of rights and users may be assigned to roles according to their functions.

Specific policies

Some systems require more specific policies; for example, the military put high emphasis on secrecy.

Confidentiality policies

Document classification—Documents are classified according to the sensitivity of their information.

People are given clearances. The policy defines a relationship between classification and clearances. For example, the clearances and the classifications may be hierarchical levels: Top Secret, Secret, Confidential, Public, and a user cleared for a given level can read all the documents at her level or below.

Categories—They define vertical partitions of the levels, e.g., Army, Navy. Now, not only the classification must be appropriate to read a document but also the user category must match or include the category of the document.

Integrity policies

Authorized actions—People can perform only actions for which they are authorized.

Rotation of duty—A task should not always be carried out by the same person

Operation sequencing—The steps of some task should be carried out in a specific order

Constrained change—Data can be changed only in prescribed ways

Policy combinations

Chinese Wall policy—Information is grouped into “conflict of interest” classes and a person is allowed to access at most one set of information in each class.

Originator controlled (ORCON)—A document is released only to people or units in a list specified by the originator.

System policies.

Policies can refer to low-level system aspects. For example, a User Account/ Password policy [And02], defines aspects such as the length of passwords, what characters they may or may not have, and how often they should be changed. A common error is to define low-level policies without using higher-level policies as a reference. For example, Visa requires that online merchants using their cards should: install a firewall, keep security patches up-to-date, encrypt stored and transmitted data, etc. These policies are too detailed to be effective and are restrictive to the participant merchants because they are not based on higher-level policies.

Some general policies for systems include:

Isolation or containment. A system should be isolated from external systems, a process should be isolated from other processes.

Controlled sharing. Resources or information should be shared by processes or systems in a controlled way, subject to specific authorizations.

Memoriless systems. A program should not keep any traces of its past executions.

An example of policies

The following is a possible set of policies for a university system, assuming also a closed system policy:

An instructor can look at all the information about the course he is teaching.

An instructor can change the grades of the students in the course he is teaching

A student may look at her grades in a course she is taking or has taken.

The department head can add/delete course offerings

The department head can add/delete students from course offerings and can assign instructors to them with the agreement of the corresponding faculty.

Faculty members can look at information about themselves

A department chairman can look at information about his department and can change information about faculty and courses

A dean can look at the information of his college

Use of roles in policies

It is important to define roles with respect to the information produced or used in an institution or system. Some possible roles with respect to documents are:

?? Originator—The person who issues a document

?? Authorizer—The person who controls access over the document

?? Custodian—The person who keeps the document and controls its use

?? User—The person who reads or modifies the document

?? Auditor—The person who checks the actions, results, and controls

We can also define roles for people according to their job functions and assign rights according to these functions; for example, manager, secretary, student, instructor. This is the basis of Role-Based Access Control (Chapter 4).

Policies and secure systems design

Once we have a list of the threats to our system we can decide which ones of these threats are important and how we can stop them, according with the policies of the institution; that is, the policies will guide the selection of the specific mechanisms we need to stop

the threats. For example, if secrecy is important we must protect against viruses or Trojan Horses that may compromise secrecy. Policies are also important for evaluating a secure system, if a system fulfills its policies, it is secure for our purposes.

The security policies should be reflected in the security mechanisms used in the different architectural levels. The lower level mechanisms should enforce the policies defined at the high levels. Most commercial systems do not apply the policies described above; for example, in Unix a file creator becomes its administrator and user, which violates the policy of separation of duty. We will discuss specific policies when we discuss specific architectural levels. Some books try to enumerate security policies [Woo00], although they consider any common sense recommendation a policy.

When we have hierarchies of policies we can have conflicts and it is important to resolve them before continuing with the more detailed design.

At this moment we can consider the regular use cases of the system to define the rights that users must have to be able to perform their functions [Fer97].

Some security policies can be represented by formal or semi-formal models. A model allows us to analyze security properties and is the basis for system design. The next chapter surveys some common security models.

References

[And02] M. Address, “An overview of security policies”,
<http://searchsecurity.techtarget.com>, December, 2002.

[Bla00] D. Blacharski, “Emerging Technology: Create order with a strong security policy”, *Network Magazine*, July 2000,
<http://www.networkmagazine.com/article/NMG20000710S0015>

[Fer97] E.B.Fernandez and J.C.Hawkins, “Determining role rights from use cases”,
Procs. 2nd ACM Workshop on Role-Based Access Control, November 1997, 121-125.

[Mea00] N. Mead (coordinator), Malicious IT Roundtable: Roundtable on Information Security Policy, IEEE Software, July 2000. Also in:
http://www.computer.org/software/so2000/pdf/s5mt_rt_web.pdf

[Pol] Policy 200X: Workshop on Policies for Distributed Systems and Networks,
<http://www-dse.doc.ic.ac.uk/events>

[Sum97] R.C.Summers, *Secure computing: Threats and safeguards*, McGraw-Hill, 1997.

[Woo80] C.Wood, E.B.Fernandez, and R.C. Summers, “Data base security: requirements, policies, and models”, *IBM Systems Journal*, vol. 19, No 2, 1980, 229-252.

[Woo00] C.C. Wood, *Information security policies made easy*, Version 7, 2000.
<http://www.pentasafer.com/products/vsapolicybook.htm>

