



# Information Access Policy Management

## Problem

The Jericho Forum has been described in a crude way as ‘those people who want to get rid of firewalls’. That’s not strictly accurate – Jericho believes that protection should be applied close to data (JFC#1, JFC#9)<sup>1</sup>, and that firewalls should just be ‘quality of service separators’.

The Jericho Forum philosophy leads to access policies evolving from coarse, infrastructure-oriented statements to fine-grained business-oriented ones. But there’s a problem – there are an awful lot more data items than there are firewalls. If each data item has to be individually protected, how are we going to manage the enormous number of information access policies we will need?

## Why should I care?

- Efficient information access policy management is critical to securing an agile, rapidly changing enterprise.
- The problem of how to specify, apply and enforce information access policies on mobile data is critical if privacy and intellectual property rights issues are to be properly addressed. In particular, eBusiness relies on free flow of information, but information owners need confidence that their information will be properly handled by the holders.

## Recommendation/response

Jericho Forum believes that:

- Information access policies must be expressible in powerful languages that can accurately capture the intention of the creator.
- Secure systems need to separate out the administration points, decision points and enforcement points for information access policies.
- Businesses need to adopt new techniques for understanding their security imperatives so they can be accurately encoded.
- A set of interoperable global identifiers needs to be developed.
- Where organisations exchange data, they should also expect to exchange information access policies covering how that data should be handled.

---

<sup>1</sup> The term JFC#n refers to the relevant Jericho Forum Commandment number – see [http://www.opengroup.org/jericho/commandments\\_v1.2.pdf](http://www.opengroup.org/jericho/commandments_v1.2.pdf)

## Background & rationale

### What is an Information Access Policy?

A security policy is a rule that an organisation must follow in order to meet its security objectives. An Information Access Policy is a particular type of security policy specifically related to the security of information and its underlying data. There are many types of information access policy, some examples are given below:

- ‘Personnel information shall be readable by the subject, the HR department, and the subject’s manager’. This is a business-oriented human-readable policy. It is generic, in that it can be applied to many different assets.
- A file ACL is a machine-readable policy. It is infrastructure-oriented and applies to just a single object.
- ‘Users must not install applications on their laptops without permission from the security department’. This looks like the previous business-oriented policy, but note that it does not actually say what applications are permitted and which are not, nor does it even give criteria for permitting an application. Rather, it describes a procedure for obtaining permission (‘ask the security department’).
- A software or DRM license is a form of policy. This is a machine-readable business-oriented policy and may specify controls over copying and expiry.

It should be clear that policy statements are critically important in controlling organisations and computer systems.

We can see a dichotomy above between human-readable and machine readable policies. IT systems, of course, run off machine readable policies, but these are hard for humans to understand. Human-readable policies, on the other hand, cannot be understood by computers.

Another distinction above is between specific information access policies that cover a single asset, and generic policies that cover whole classes of asset. Many machine-readable policies today are specific. That means that a separate policy must be written for every asset, even if it takes a similar or identical form to a previous policy. It becomes very difficult to check that all policies in a class are correct, and it is very difficult to keep policies in step with changing requirements. Many organisations will have millions, or even billions, of data items, and it is not practical to devise a separate policy for each one.

Finally, the machine readable policies above are expressed using very basic language; for instance, a file ACL is expressed using user identities, group memberships, and the ‘or’ operation. It is not capable, therefore, of encoding many of the business oriented policies given above. This lack of expressive power makes it hard to accurately reflect business requirements, and hard to keep policies up to date as infrastructures change and as data becomes more mobile. Such requirements tend to be implemented by dedicated program code, and thereby become hard to understand, test and maintain. A fine grained policy needs to be much more accurate than a coarse-grained one.

### What is hard about information access policies?

Most organisations have no idea what their information access policies are – even (especially) those which claim to have a documented security policy.

More than that, the question of how to relate an information access policy to the business imperatives that justify it is only just starting to be asked. This makes it hard to know how to create a business-oriented information access policy.

Most real-world policies need to cope with numerous exception conditions. For instance, a doctor should not see medical records without the subject's consent, but if the subject is wounded and unconscious he cannot give consent so there will need to be a legitimate way to bypass the information access policy – audited two man rule, for instance.

Many information-based services use data from many different providers, who have to give this data to a customer or supplier in order for the service to function. Once the data is handed over, the owner loses control over it.

The data owner has to be able to:

- Set an information access policy about how his data should be handled.
- Have some confidence that his data will be handled according to the information access policy.
- Have confidence that the information access policy is bound to the data for the life of the data, including copies of data.

The situation can quickly become very complex, with most real-world services using data from many different providers, each with its own information access policy requirements, processed through several stages, each of which may change the requirement.

## Key Challenges and Next Steps

Information access policy requires organisational change as well as technical innovation.

- Organisations need to understand what their information access policies really are. It is important to realise that there will be many governance patterns for policies, e.g.
  - Automated control. The owner of a data item specifies an information access policy about how it may be used. All holders of the item must consult the policy before they may give access to it.
  - Workflow-based control. In this case it is not possible to specify a simple information access policy that a machine can follow, so the data owner (or his delegate) must be involved personally in the authorisation process.
  - Accountability. In this case the data owner trusts a data holder to control access to his data, but retains the right to know who has accessed his data and why, and to hold accessors accountable for their access.
  - Time-limited permissions. The data owner gives permission for a very short period of time after which the data holder must seek permission again.
- To allow such complex models to operate, systems must be able to separate information access policy administration (which will be done by the data owner); policy decision (which will be done by the owner or his delegate); and policy enforcement (which must be done by all data holders). This is the basis of standards such as XACML.
- Information access policies need to become more sophisticated than ACLs. Essentially they are specialised programs. Experiments have been conducted into expressing policies as proof obligations, for example. XACML is a valuable step in this direction, but only a step.
- Means of managing identities between different attribute authorities. The Open Group's Common Core Identity white paper addresses this issue.