Realtime publishers

The Essentials Series: Enterprise Identity and Access Management

# Compliance

*sponsored by*

QUEST SOFTWARE®

by Richard Siddaway

## Copyright Statement

# Compliance

Compliance is playing an increasing part in the administrator's life. Increased levels of regulation mean that organizations have to be able to prove who can access what data and why that access was granted. When the access was granted and by whom will also need to be recorded.

## Auditing

Auditing is how you prove that the correct things are happening and, sometimes more importantly, detect that incorrect things are happening.

### *Access*

Auditing access falls into two areas:

- Who is logging on to my network and when?
- What are they accessing?

In AD, the information about logon is stored in the event logs on the Domain Controller that processes the logon request. The event logs of all Domain Controllers must be examined to generate a full picture of activity.

The event log information should be collated and stored for future analysis. At the very least, checks should be made for authentication attempts with multiple failures, authentication events that occur at unexpected times, or authentication against accounts where the user is known to be absent.

Auditing access to data can again mean collating information from disparate systems such as the file server event logs and the database audit system. Deciding which data needs to be audited is the key to this process. Auditing access attempts that have failed on confidential data is as important, if not more important, than auditing successful access attempts.

In a heterogeneous world, non-Windows machines will significantly complicate this picture. The logging mechanisms are different for non-Windows systems and they are all different to Windows. This leads to a situations where

- Different information is being logged
- Different and incompatible log formats are used
- The information cannot be correlated across systems
- Actions cannot be tracked that cross system boundaries

This will have highly detrimental effects on any analysis of the collected auditing information.

### *Privilege Use*

Administrators have a high level of privileged access to the systems. Auditing their activity can become a necessity. Why and when was an individual account changed, and by whom? Was the change part of a documented process or as a response to a Help desk call? Changes to group membership can have significant impact on the accessibility of data. Who added that user to the group? Monitoring privilege use in this way may seem overly intrusive to many administrators, but it should be viewed as a protection mechanism. This may even extend to situations where keystrokes are logged for possible playback during forensic analysis. One scenario that comes to mind is the detection of tampering with event logs.

### *Separation of Duties*

One very important point regards the separation of duties between those administering the system and those auditing the system. The auditors should not be responsible for administration and those performing the administration should not be responsible for producing the audit reports. Is this possible in a heterogeneous system or do some systems allow only high-level administrators to produce these reports?

### *People*

Administrators are in a position with a high level of trust. Do you trust your administrators? If not, why are they administrators! What checks do you perform when hiring administrators? Trustworthy administrators are one of the cornerstones of a sound identity and access management strategy.

## Reporting

Generating reports from your authentication and authorization systems is a task that is often required in the compliance process. Active Directory (AD) does not have any native reporting tools. Other directory systems may have the ability to generate the required reports but most do not.

Reports can be generated from a single directory system, though it may mean the administrators having to learn scripting techniques. How can this be extended to multiple systems or platforms? Even if the required scripts can be generated to access the multiple directory systems, how will the information be collated? How can you prove that a particular userid on two disparate systems relates to the same real individual?

The reporting of what systems and data a particular individual can access has even more issues. Windows holds the permissions on the file object. How are you going to report on every file on your file servers? A well-designed folder and authorization system will help resolve this issue. Design the folder structure at the same time as the group structure so that permissions are granted as far up the folder tree as possible.

Unfortunately, in many organizations, unstructured data means just that. Resolving the tangle of permissions and groups can take a very large amount of effort.

Heterogeneous systems bring massive levels of complexity and difficulty to reporting. The issues around different log formats and information in auditing systems, discussed earlier, causes great difficulty in reporting:

- Do the people producing the reports understand the different formats and information types?
- Can the auditing information be easily found?
- Do the report generators have permissions to access the data?
- Can the multiple reporting tools produce compatible output?
- Is the data encoding and collation compatible across the reporting tools and mechanisms?

These issues will grow exponentially as the number of different system types increases.

## Proving Compliance

The auditing and reporting that has been discussed goes a long way to proving compliance, but to ensure a robust solution, the activity must be viewed as a business process with a set of well-documented steps. Not only can you prove you are compliant but you can prove the process you use!

Ideally, at least some of the compliance activity will occur in real time. If a login attempt is made on a normal user account well outside business hours, this should be reported and the appropriate alerts issued.

## Solution

An authentication and authorization strategy should be developed as part of the enterprise's architecture. The aim of the architecture should be to simplify the authentication and authorization mechanisms used within the organization. The ideal solution, for an organization with a significant Windows infrastructure, is to consolidate their identity and access management for heterogeneous systems into AD.

### *Consolidation Methodologies*

Consolidating the multiple, disparate non-Windows directories and identities within an organization must be viewed as a prime requirement. This will include AD, Unix, Linux, Mac, and Java and other applications. AD is the logical contender for the consolidation target:

- It is already in use within your environment and is required for applications such as Exchange
- Administration expertise already exists
- No software costs—part of Windows
- It can be extended
- Interaction with other systems is possible because of the use of standards such as Kerberos and LDAP in AD and PAM and NSS in Unix/Linux, or pluggable authentication such as GSSAPI in some applications.

The main issue with directory consolidation is the actual migration. In many cases, it is not possible to simply create new user accounts in the target directory as the identifiers, such as SID, will change so permissions will not be carried across. It is preferable to use a tool to perform the migration. Doing so ensures that the correct information is replicated into the target directory and that permissions can be preserved. This is true even if the accounts are being migrated from an existing AD.

One common scenario is where an organization has grown through acquisition. The resultant organization contains a number of directories. There is no single view of the organization and no single administration group. This leaves identity management as a rather ad hoc process with no consistency or standards applied across the organization. By consolidating to a single AD, it becomes possible to easily apply the same standards across the organization. This can have other benefits as applications such as email can be consolidated, which will further simplify administration.

### *AD*

Direct AD-based authentication and authorization for Unix\Linux, Apple, and Java can be achieved; however, other systems— such as enterprise applications, mainframes, and so on—can be integrated with AD in a number of ways and to varying degrees of complexity:

- Use tools for basic password synchronization. These tools are not necessarily easy to use and only provide a small part of the required solution.
- Perform a manual install of Kerberos and the other required utilities; join the non-Windows machine to the domain after manually configuring the authentication protocols and modules; and create the required accounts in AD
- Use a commercial toolset to make non-Windows systems look and act like Windows clients so that they can authenticate against an AD account. The toolset includes the use of Kerberos and can bring other benefits such as participating in the AD time synchronization regime or Group Policy.

The third option is preferred as it eliminates the need for manual configuration. This is especially important if a number of non-Windows machines are to be incorporated into AD. The effort required to perform this manually is not trivial.

The migration of the accounts into AD lays the foundation for a consolidation of identities within the organization with the ultimate goal of providing a single identity that can be used to authenticate to all the systems and applications in the enterprise.

Authorization can also be managed from AD as the group information is migrated from the separate systems into AD. It is possible to delegate the administration of users and groups that are just used on non-Windows systems, if required. A natural byproduct of this consolidation is creating a single point of audit that crosses platform boundaries and can leverage well-established and proven AD-based audit and reporting tools.

## Extending the Reach of AD

AD is viewed as providing the authentication and authorization functionality for Windows. After consolidating the Windows-based directories within AD, it is possible to extend the reach of AD to non-Windows–based systems. A number of non-Windows systems including Apple and Unix\Linux can utilize AD for authentication. AD can be the source of login automation with application and mainframe systems to provide a single sign-on (SSO) experience to the user population.

It is also possible to extend the reach of AD to cover applications and Web access either internally or to external parties. Applications and services can be made available to business partners with authentication and authorization managed by AD federation—extended to non-Windows applications and systems.