Realtime
publishers

The Essentials Series: Enterprise Identity and Access Management

Authentication

sponsored by

QUEST SOFTWARE®

by Richard Siddaway

Copyright Statement

Authentication

In a Windows-based infrastructure, Active Directory (AD) is the basis of identity and access management. It provides the data store against which users are authenticated and holds information, in the form of group membership that is used to authorize access to network resources. However, the vast majority of organizations have a number of non-Windows systems and applications that face similar identity and access management challenges.

This series will examine the issues surrounding AD-based identity and access management and covers the following topics:

- Authentication—Proving who you are
- Authorization—Controlling what the authenticated person can access
- Administration—Challenges of administering the actual life cycle of the identity
- Compliance—Proving the identity and access policy is secure and working

Identity and access management starts with identity. Who are you and how can you prove it? Authentication is the act of proving you are who you say you are. It is accomplished based on one of three broad techniques:

- Something you know. In most cases a userid and password.
- Something you have. An example would a hardware token used for VPN access.
- Something you are. The use of biometric techniques for authentication would fit in this category.

The majority of organizations are using the “something you know” technique in which users enter a userid and password to log on to the network.

Issues in Authentication

In theory, authentication is very straightforward to manage. An administrator creates a userid for the user. The user creates a password according to the company policy, and the user can then authenticate. Unfortunately, there are a number of issues with the use of passwords.

Passwords—The Weakest Link?

Passwords are required to authenticate to the network but are often regarded as a necessary evil by administrators and users. There is a conflict between the need to use stronger passwords by increasing their length and complexity and the ease of use for the user population. Passwords cause a significant proportion of the Help desk calls within an organization. Increase the length and complexity of passwords, and those calls rise dramatically, which increases your cost of operations. Make the passwords too weak, and the organization's security is compromised.

A password that is too complex will be difficult to remember. This can cause passwords to be written down and left where they can be found. Passwords can become predictable because users just change the last digit of the password or they base the password on a potentially well-known fact, such as their pet's name. Users freely share their userids and passwords, so you cannot be sure that the userid accessing the network is the user assigned to that userid.

In heterogeneous environments, users may often have multiple passwords they need to remember in order to access applications and resources on the network. This complexity raises operational costs and causes problems for the user population.

Privileged Accounts and Service Accounts

Administrator and other privileged accounts, such as the Unix root account or an administrator account for an application such as Exchange or SQL Server, need more security to prevent unauthorized access. They are a prime candidate for the multiple password policies functionality in Windows Server 2008 AD but similar functionality rarely exists in non-Windows platforms such as Unix/Linux. As these accounts have the ability to change password policies, periodic checks must be performed to ensure the policies are still being observed.

Applications such as Exchange and SQL Server run as a service. These services need an account they can use for authentication when they start. This account is often a domain account rather than a local account. Some Windows services (for example, the cluster service) require a domain account as well. How should the passwords used by these accounts be managed? The easiest way is to set the password so that it does not expire. However, these passwords become known, and when administrators leave an organization, these known passwords can create a security hole. However, trying to change these passwords on a regular basis is problematic as administrators need to manually apply the change and ensure that the password does not expire.

Heterogeneous Environments

It is rare for an organization to have a single platform. Many, if not most, organizations have mixed environments with Windows, Unix, Linux, Apple, and possibly even a mainframe. Each environment, and often each box in the environment, has its own authentication mechanism, so users will have multiple passwords and userids. This increases complexity for the users and the administrators.

Remote Users

Authenticating remote users against the corporate network raises further authentication issues:

- Is authentication passed through to AD?
- Will a separate authentication mechanism be required?
- Will all access be via VPN or will remote access technologies such as Outlook Anywhere or Outlook Web Access be used?
- How many systems and applications will need to be accessed, and do they require unique userids and passwords?

Authentication Methods

A number of authentication mechanisms can exist within an enterprise.

Kerberos

Kerberos in conjunction with LDAP provides authentication in AD. Non-Windows environments do not use Kerberos for authentication although some may be "Kerberos-aware". Solutions exist that can "Kerberize" non-Windows systems to allow them to participate in the AD Kerberos authentication trusted realm.

Certificates

Certificates are a form of "what you have" authentication. The certificate is validated against a certificate authority, and if it is valid, the authentication succeeds and authorization can proceed. The certificates are mapped to accounts in a directory such as AD. The drawback to using certificates is the need to maintain a PKI infrastructure to issue and administer the certificates. This requires trained personnel capable of fulfilling these administrative tasks. If an organization maintains its own PKI infrastructure, it is not an easy task to issue certificates to members of external organizations that you might want to authenticate to your network. The alternative is to obtain certificates from a commercial certificate authority; however, the cost and administration required may prove prohibitive.

Tokens

Tokens, either hardware- or software-based, are another example of “what you have” authentication. Commonly used for VPN access they work well from the user perspective in that there is one less password to remember. Token based access has a number of issues:

- Administrative effort required to issue tokens
- Increased complexity caused by adding another authentication system into the environment
- Overhead of administering the token system
- Cost of acquisition and ongoing operations

The apparent gains at the user side can be easily outweighed by these disadvantages to the organization.

Smart Cards

Smart cards supply authentication data from a certificate on the card. It is usually necessary to also supply a PIN. By supplying “something you have” (that is, the card) and “something you know” (that is, the PIN), you have a more secure authentication environment.

However, the issuing and administration of the cards may become problematic. The cost of supplying the cards and the associated readers may become prohibitive. There are a number of questions that need to be resolved:

- How do your users prove their identities in order to be issued a card?
- How are lost cards dealt with?
- Is there a mechanism for providing temporary cards?
- What happens in the situation in which a card cannot be used?
- Can your smart card infrastructure support multiple platforms and applications?

Smart cards offer great potential, but there are many administrative details to be resolved.

Biometrics

In theory, biometric-based authentication should be the ideal solution. Biometric data such as fingerprints or retina scans can be used to establish identity with a greater degree of certainty than with other methods. However, biometric-based authentication cannot be viewed as a mature technology.

The required technology is not necessarily available for every machine in the enterprise. The biometric data needs to be recorded and associated with a particular individual. It also needs to be held in a secure fashion that prevents replay attacks. Biometric-based authentication techniques have not gained wide acceptance with many users who are reluctant to adopt them.

Proprietary and "In-house"

Many applications that are obtained from a third party or written in-house may adopt their own authentication mechanisms. This often requires the user to enter an identity and password into the application. The credentials are checked against data stored within a database controlled by the application.

This type of functionality should be discouraged. Development teams, especially in-house teams, should be encouraged to use AD for authentication. If it is not possible to use AD, because of schema change issues, for example, then consider the use of Active Directory Lightweight Services (AD LDS, formerly ADAM). The AD LDS instance can be populated with identity data from AD. In this type, the database table containing the security information can be accessed by database administrators. This could provide a security loophole for an unscrupulous administrator to exploit.

Single Sign-On

Single sign-on (SSO) is the ultimate goal within an authentication strategy. This enables the user to access all data and applications via a single identity and authentication event. SSO can be accomplished by performing password synchronization between matched identities in multiple stores or replaying the authentication event across other directory stores. The preferred approach to SSO is for the authentication information to be held in a single consolidated directory so that a single authentication mechanism is used across the enterprise.

Summary

Identity and access management is a cornerstone of security for the enterprise. If you do not know who is accessing your environment and cannot control who is working with your resources, you cannot have a secure environment. A crucial piece of this security is the ability to verify that users are who they say they are through consolidated and strong authentication techniques.