Global Information Assurance Certification Paper

Not Another ID and Password

A Look at Single Sign-On

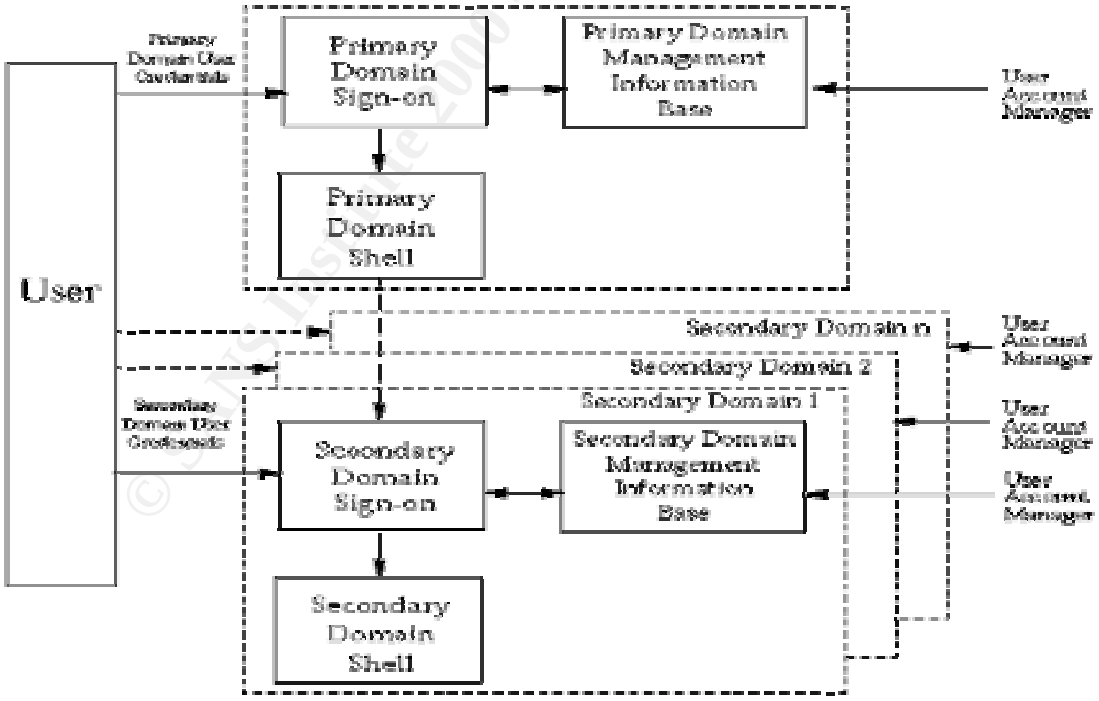
Michael B. Bonham, II

Multiple accounts and passwords are cumbersome and time consuming for network administrators and users alike. Single Sign-On (SSO) technology allows the user to authenticate once and still access additional network resources without having to re-authenticate. SSO helps to reduce user frustration and administration time.

Problems with Multiple Authentication

A typical user has multiple accounts and passwords that access different platforms and applications across heterogeneous environments. The user enters their account and password to sign on to the primary domain from their workstation. Then, using the operating system shell, they are able to access their secondary domains and applications.

However, in order to access a secondary domain the user has to perform a secondary domain sign-on and provide another set of credentials. The user has to repeat this separate sign-on for *each* secondary domain or application they want to access. This process is illustrated in the figure below:



Users often complain that keeping track of so many different accounts and passwords is difficult as well as time-consuming. There are a couple of methods a typical user will try in order to avoid the burden of multiple account information.

The first is for the user to commit their different account information to paper and store it under the keyboard, or tape it to their monitor. The second is to use the same account information for each system they need to access. As you can imagine, the first method creates a huge security risk. The second method might work for a time, but because systems have different security policies, the passwords would expire at different times, generating additional system administration tasks.

Multiple password resets and account lockouts spawn another set of problems with multiple system authentications. For instance, user administration quickly becomes a burden, which takes a good portion of time from the help desk and system administrators. More than fifty percent of all help desk calls are password resets and account lockouts, which increases support cost and time.

In addition, users experience delays in obtaining access to systems and resources because the System Administrator has to perform several tasks just to add one user to multiple domains:

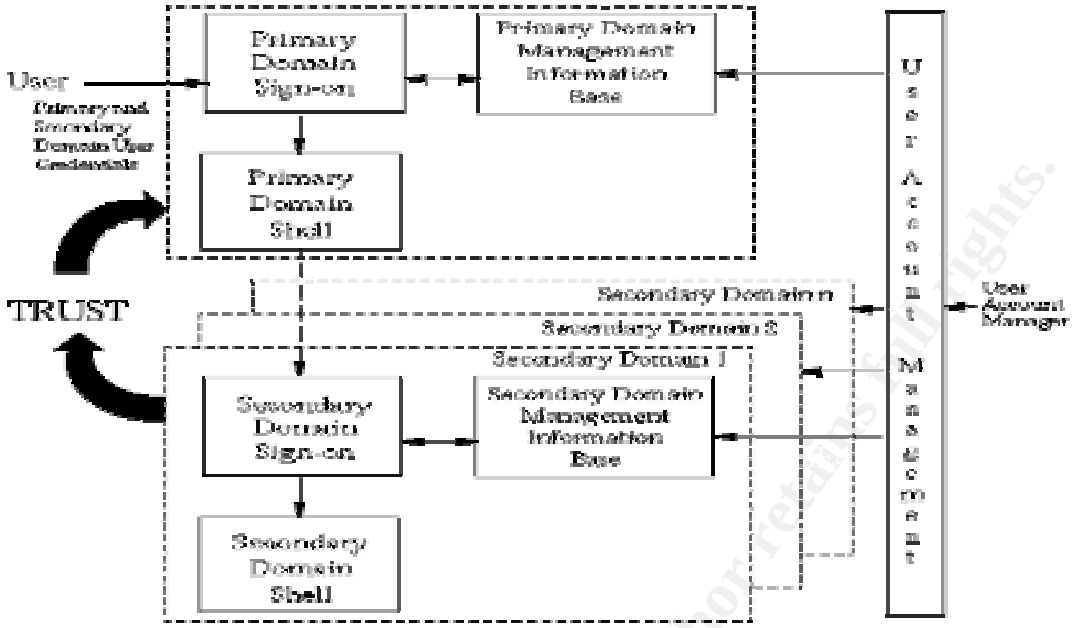
- Confirm which domains to grant access to
- Create the account on each domain
- Assign proper access control list to each account

Security suffers from an account deletion request for the very same reason. Because of the multiple systems that are supported, it is possible for the System Administrator to overlook removing the account from one of them. This creates another security risk and possibly provides unauthorized access into systems.

Single Sign-On (SSO)

Single Sign-On provides the users with a one-time sign-on. SSO reduces help desk support cost, simplifies system administration tasks, and decreases user down time.

On a SSO system, the primary domain sign-on will collect all necessary user identification and credential information needed to authenticate to all other secondary domains. That information is then used to transparently authenticate to each secondary domain. The illustration below displays this process:



There are several ways the credentials can be passed from the primary domain sign-on to the secondary domain sign-on: directly, indirectly, immediately, or stored in cache. Credentials that are passed directly are sent straight from the primary domain directly to the secondary domain. Credentials are passed indirectly when the user's primary sign-on retrieves a new set of credentials from a single sign-on authentication server and then supplies the new credentials to the secondary domain sign-on. When credentials are passed immediately connections are established for all secondary domains as part of the initial sign-on to the primary domain. Credentials can also be stored in cache and pulled from cache at the time a connection to a secondary domain is requested.

Simple SSO

A simple version of Single Sign-On stores all login credentials on an authentication server. Through the SSO client the user signs on to the authentication server from their workstation. The authentication server instructs the client which resources are available to that user. When the user attempts to accesses a secondary resource the client then uses the credentials supplied by the authentication server to establish a connection. Password security is still maintained because the authentication server can maintain complex and different passwords for each resource.

The simple version of SSO will help alleviate help desk support cost but does not help reduce system administration tasks. User accounts and access lists still have to be maintained for each of the different resources.

Advanced SSO

In an advanced version of Single Sign-On the user signs on to the workstation client with their SSO login credentials. The client will send the users credentials in encrypted form to the authentication server. After receiving the credentials the authentication server will validate them and then will let the client know which resources are available to the user based on current security policies and roles. The user is then presented a graphical interface that displays all their available resources. The user will then be able to select a resource from their desktop and be authenticated to it transparently.

With centralized security administration and tighter security controls, advanced SSO addresses several security risks:

- Provide token-based authentication between the client and authentication server. A non-forgeable, non-reusable token can be created that identifies the user to authorized resources. Kerberos and Keon are two of the most used token-based systems found in SSO implementations.

- A centralized security administration utility, which allows accounts to be administered across different resources for role-based scenarios. With this utility accounts that have been requested for removal can quickly be deleted across all resources. This helps to strengthen security by eliminating the human element that might miss removing the account from one of the domains.

- Centralized administration helps to greatly reduce the time it takes to add accounts and change permissions across different resources through the use of a single interface.

- Protect passwords from interception by offering a wide range of controls and encryption. Random passwords are generated and changed on the different systems for the user. This increases security by having known strong passwords and requiring the user to authenticate only through the SSO system, in effect, disabling backdoor access.

- Auditing and alarms can be set to enhance security. Security polices are enforced because auditing makes users accountable for their activities. Alarms are set to notify when system changes are made or failures have been detected allowing administrators to be more responsive.

Security

Some security professionals are concerned with Single Sign-On. If a users account and password are compromised then the attacker has access to all the resources that user was authorized. Using a secure two-factor authentication method like one-time passwords, smart cards with PKI or biometrics would provide some controls to that risk.

SSO Summary

Single Sign-On can save time, money and frustration. Users only have one password, which reduces login time and frustration. Password resets are decreased due to the one-time authentication, which in return, reduces help desk support cost. Being able to do administration centrally improves the time it takes to add and remove accounts or modify access rights.

SSO also improves security as users will no longer have to remember as many passwords and will be less inclined to keep a written list of them. Administrators are able to confidently delete a user from all resources. Backdoors are closed because a user only knows their SSO login, preventing them from logging directly on to the resource.

Reference

The Open Group, *"Introduction to Single Sign-On"*
Illustrations used from *"Introduction to Single Sign-On"* with permission.
www.opengroup.org/security/sso/sso_intro.htm

Fred Trickey, *"Secure Single Sign-On: Fantasy or Reality?"*
www.gocsi.com/sso_ft.htm

Philip Carden, *"New Face of Single Sign-On"*
www.nwc.com/1006/1006f1.html

iD2 Corporation, *"Secure Single Sign On"*
www.id2.se/sso/main.asp