INTRODUCTION

As IT systems proliferate to support enterprise processes, users and system administrators are faced with an increasingly complicated interface to accomplish their job functions. Users typically have to sign-on to multiple systems, necessitating an equivalent number of sign-on dialogues, each of which can involve different usernames and authentication information. System administrators are faced with managing user accounts within each of the multiple systems to be accessed in a coordinated manner in order to maintain the integrity of security policy enforcement. This legacy approach to user sign-on to multiple systems is illustrated in Exhibit 1.

Historically, a distributed system has been assembled from components that act as independent security domains. These components

PAYOFF IDEA

In many enterprises, employees typically require access to many computers in the normal course of their work. In most cases, this requires that they hold usernames and passwords for each computer they need to access, and results in too many passwords for individuals to remember as they move from system to system. Consequently, stories abound of passwords being written on notes stuck to terminals and workstations. Given that users move or otherwise change jobs, the management of these user accounts is an arduous and expensive task. This article discusses how security enterprises like The Open Group; Mercury Information Technology, Inc.; Platinum Technology IP, Inc.; and Schumann Security Software, Inc., are developing an industry-wide product standard for single sign-on (SSO). This standard will enable a user to log-on once to the enterprise, instead of requiring an individual log-on to each individual system for which access is required. In this article one learns how SSO programs are used to allow a user to authenticate himself once, and from then on be able to access additional network resources without providing additional passwords.
comprise individual platforms with associated operating system and applications.

The components also act as independent domains in the sense that an end user has to identify and authenticate himself or herself independently to each of the domains with which he or she wishes to interact; this scenario is illustrated in Exhibit 1. The end user interacts initially with a primary domain to establish a session with that primary domain. This is termed the “primary domain sign-on” as shown in Exhibit 1, and requires the end user to supply a set of user credentials applicable to the primary domain (e.g., a username and password). The primary domain session is typically represented by an operating system session shell executed on the end user’s workstation within an environment representative of the end user (process attributes, environment variables, and home directory). From this primary domain session shell, the user is able to invoke the services of the other domains, such as platforms or applications.

To invoke the services of a secondary domain, an end user is required to perform a “secondary domain sign-on.” This requires the end user to supply a further set of user credentials applicable to that secondary domain. An end user has to conduct a separate sign-on dialogue with each secondary domain that the end user requires to use. The secondary domain session is typically represented by an operating system shell or an application shell, again within an environment representative of the end user. From the management perspective, the legacy approach requires independent management of each domain and the use of multiple user account management interfaces. Consideration of both usability and se-
curity gives rise to a need to coordinate and, where possible, integrate user sign-on functions and user account management functions for the multitude of different domains now found within an enterprise. A service that provides such coordination and integration can provide real cost benefits to an enterprise through:

- improved security via the enhanced ability of systems administrators to maintain the integrity of user account configuration, including the ability to inhibit or remove an individual user’s access to all system resources in a coordinated and consistent manner
- improved security via the reduced need for a user to handle and remember multiple sets of authentication information
- reduction in the time taken, and improved response, by systems administrators in adding and removing users to the system or modifying their access rights.
- reduction in the time taken by users in sign-on operations to individual domains, including reducing the possibility of such sign-on operations failing.

Such a service has been termed “single sign-on” after the end-user perception of the impact of this service. However, both the end-user and management aspects of the service are equally important. This approach is illustrated in Exhibit 2. In the single sign-on approach, the system is required to collect from the user, as part of the primary sign-on, all the identification and user credential information necessary to support the authentication of the user to each of the secondary domains that the user may potentially require to interact. The information supplied by the user is then used by single sign-on services within the primary domain to support the authentication of the end user to each of the secondary domains with which the user actually requests to interact. The information supplied by the end user as part of the primary domain sign-on procedure can be used in support of secondary domain sign-on in several ways including:

- directly: the information supplied by the user is passed to a secondary domain as part of a secondary sign-on
- immediately: to establish a session with a secondary domain as part of the initial session establishment (this implies that application clients are automatically invoked and communications established at the time of the primary sign-on operation)
- indirectly: the information supplied by the user is used to retrieve other user identification and user credential information stored within the a single sign-on management information base; the retrieved information is then used as the basis for a secondary domain sign-on operation
- temporarily stored or cached: and then used at the time a request for the secondary domain services is made by the end user.
From a management perspective, the SSO model provides a single user account management interface through which all the component domains can be managed in a coordinated and synchronized manner. Two significant security aspects or approaches to SSO systems are:

1. Authentication credentials must be protected when transferred between the primary and secondary domains against threats arising from interception or eavesdropping leading to possible masquerade attacks.

2. The secondary domains must trust the primary domain to:
   a. correctly assert the identity and authentication credentials of the end user,
   b. protect the authentication credentials used to verify the end-user identity to the secondary domain from unauthorized use.

With that in mind, this article discusses how security enterprises such as The Open Group; Mercury Information Technology, Inc.; Platinum Technology IP, Inc.; Syllogic B. V; and Schumann Security Software, Inc. are developing industry wide product standards and new technologies for single sign-on (SSO) — starting with The Open Group.

**SCOPE OF THE SINGLE SIGN-ON STANDARD**
The Open Group defines the scope of the Single Sign-On Standard (SSOS) as services in support of the development of applications to
provide a common, single end-user sign-on interface for an enterprise; and as services in support of the development of applications for the coordinated management of multiple user account management information bases maintained by an enterprise.

Functional Objectives: User Sign-on Interface
The following functional objectives have been defined for the SSOS in support of a user sign-on interface. For example, change of user-controlled authentication information shall be supported. This is interpreted as initially being restricted to change of user password, although capability for future extension shall not be precluded.

Furthermore, provision of a service to enable a caller to notify the SSOS implementation of a change of user-controlled authentication information by an application other than the SSOS implementation is an optional requirement and may be supported. Also, SSOS shall not predefine the timing of secondary sign-on operations.

In addition, support for the initiation of cleanup services on session termination, or sign-off, shall be supported. Also, support shall be provided for a caller to establish a default user profile. User selection from a set of available user profiles is not required to be supported, but shall not be precluded as a future extension. Finally, the interface shall be independent of the type of authentication information handled.\(^2\)

User Account Management Interface
The following functional objectives have been defined for the SSOS in support of a user account management interface:

- creation, deletion, and modification of user accounts shall be supported
- setting of attributes for individual user accounts shall be supported
- attributes to be supported shall include as a minimum those necessary to support the SSOS.

Nonfunctional Objectives
The nonfunctional objectives of the SSOS are, for example, that the SSOS shall be authentication technology independent. The interface shall not prescribe the use of a specific authentication technology, nor preclude the use of any appropriate authentication technology.\(^3\)

SSOS shall be independent of platform or operating system. Also, SSOS shall not preclude the integration of common desktops or common servers, including mainframes. Finally, there is no expectation that such desktops or servers shall be capable of integration within SSOS without modification.
Security Objectives

There are many security objectives to be met by an implementation of SSOS. These objectives are:

- An SSOS implementation shall audit all security relevant events that occur within the context of the SSOS.
- An SSOS implementation shall protect all security relevant information supplied to or generated by the SSOS implementation such that other services may adequately trust the integrity and origin of all security information provided to them as part of a secondary sign-on operation.
- SSO shall not adversely affect the resilience of the system within which it is deployed.
- SSOS shall not adversely impact the availability of any individual system service.
- SSOS shall not provide access by principals to user account information to which they would not be permitted access within the controlling security domain for that information.
- The SSOS shall provide protection to security-relevant information when exchanged between its own constituent components and between those components and other services.

Out of Scope

Finally, there are also many aspects that are not considered to be within the current scope of SSOS. These aspects include:

- configuration and management of alternative sets of user profiles
- graphical and command line user interfaces to SSOS-based services; these are the province of applications written to utilize the SSOS
- maintenance of the integrity of the single sign-on user account information base with underlying individual service user account information bases when those underlying user account information bases are modified by means other than SSOS-provided functionality
- selection of alternative user profiles on user sign-on.
- support for single sign-on across enterprise system boundaries
- user-initiated change of non-user configured authentication information (e.g., magnetic badges, smart cards, etc.)

Now, a brief discussion of how Mercury Information Technology, Inc.² is developing an industry wide product standard and technology for single sign-on (SSO). This part of the article briefly defines two authentication-related technologies developed by Mercury Information Technology, Inc.: password synchronization and single sign-on. It describes how they each fit into the process of user authentication, and
their individual strengths and weaknesses. But first, take a look at shared systems.

**SHARED SYSTEMS**

Mercury Information Technology, Inc., defines enterprise networks as including servers that provide the following services to multiple users:

- database management
- electronic messaging
- file sharing
- printer sharing
- remove access
- run centralized applications

Most of these systems require users to identify themselves before servicing their requests.

**Authentication**

Shared systems on a network are either open to all users, or are restricted to access by authorized users only. When systems are restricted to authorized use only, they normally determine whether a given user can access a specific function based on that user's identity.

For access control based on user identity to be effective, users must be reliably identified. Authentication is the process of identifying users in a manner that makes it difficult for one user to impersonate another.

A number of technologies are available for user authentication. The most popular authentication systems include:

- biometric devices (fingerprints, retina scans, head scans, etc.)
- secret passwords
- smart cards

**Password Synchronization**

Most shared systems use passwords to authenticate users. In a network where users access multiple shared resources, they must remember a password for every system they access. Users frequently forget their passwords, and must ask the enterprise help desk to reset their passwords. Also, password synchronization software is used to ensure that each of a user's multiple passwords is set to the same value, so that users need not remember multiple passwords for multiple systems.

**Advantages of Password Synchronization.** Password synchronization is characterized by the following advantages:
• **Improved security:** A user must still enter his or her password to access a shared system. If a workstation is left unattended, only the systems currently being accessed are vulnerable to access by someone who can reach the workstation. By implementing a single point from which passwords are changed, it is possible to apply stringent requirements for what constitutes a valid password, and thus make passwords much more difficult to guess.

• **Less intrusive:** Password synchronization does not require any new servers on the network. Furthermore, password synchronization can be implemented without installing any new software on existing servers.

• **Lower cost:** Finally, password synchronization can be implemented for about one tenth the cost of single sign-on technology.

**Single Sign-on**

Single sign-on programs are used to allow a user to authenticate himself once, and from then on be able to access additional network resources without providing additional passwords. In practice, most single sign-on systems operate as follows.

First of all, the user provides a userID and password to a primary login program, which authenticates the user against a *master* system. Once authenticated, the user can request access to additional systems. When he or she does so, the single sign-on system retrieves the user's password for the new system, and starts a session with the new system using that password.

**Advantages of Single Sign-on.** On the other hand, SSO is characterized by the advantages of convenience and centralized administration. The advantages include:

• **Convenience:** Using single sign-on, users only have to type their password once when they first log in. Users can also access additional systems by just pressing a button — without entering their ID and password again.

• **Centralized administration:** Some single sign-on systems are built around a unified server administration system. These systems, allow a single administrator to add and delete accounts across the entire network from one user interface.

Platinum Technology IP, Inc., is developing an industry-wide product standard and technology for single sign-on (SSO). Take a look at Platinum Technology IP, Inc.'s SSO solution: ProVision AutoSecure Single Sign On (SSO) remote log-on and password protection, as well as other authentication techniques and how to use an enterprise SSO.
AN SSO ENTERPRISE SOLUTION

The benefits of the client/server revolution have brought a new generation of challenges and security risks. The Platinum SSO solution (like many of the SSO solutions just discussed) addresses the shift toward distributed computing that has dispersed information traditionally stored in a central mainframe into a network of interconnecting LANs — with multi-vendor servers, many different applications and services, complex log-in routines, and a multitude of passwords. Platinum’s Pro-Vision AutoSecure Single Sign-On (SSO), further addresses these problems by:

- enforcing password security policy by managing the password change process and ensuring that content rules are maintained
- increasing user productivity by allowing users to access all their authorized applications and services with a single user name and password — without having to log into each individually
- protecting information by authenticating all users and authorizing which applications and services they can use
- reducing administrator and help-desk workloads by cutting the number of passwords a user needs — and by logging users transparently into the applications and services they need to use
- reducing the system administrator’s workload by providing facilities that allow an unskilled administrator to set up and delete user accounts on different platforms from a single point

Using an Enterprise SSO

With an enterprise SSO, the user signs on to the SSO client using his or her SSO user name, SSO password, and (optionally) a role. The SSO client sends the information (in encrypted form) to that user's security server, where it is validated against defined security policies. Successful authentication means that the user's credentials are acceptable and that the user is allowed that role from that particular workstation.

Following this, the user is presented with a desktop containing all the applications which, in that role, he or she is authorized to use. Desktops are constructed in user-friendly GUI or text format, depending on the type of terminal or workstation being used.

The user can then select an application or service from the desktop and be transparently logged into that application or service by the Service Access Procedures (SAPs). No further authentication is needed. Users can also, optionally, be logged on to network servers.

If authentication fails, the enterprise SSO prevents the user from gaining access to the enterprise's applications and services. The defined security policy determines whether or not the user is given an explanation for the failure. After a predetermined number of consecutive failed log-on
attempts by the user, the workstation can be disabled pending specific intervention by the local administrator.

**Remote Log-on**

An enterprise SSO offers its entire set of capabilities to mobile users who can log on to the network via a modem from remote locations. They should be presented with their own desktops, enabling them to use their usual set of applications.

**Password Protection**

To protect its passwords from interception and misuse, an enterprise SSO offers a range of controls and encryption techniques. Where security requirements are less stringent, the requirement for password authentication can be overridden by defining roles whose users can simply log on through a user name.

**Enforcing Password Change Policy**

An enterprise SSO provides options as to how password changes are handled. The system can be set up to allow users to select the passwords to their applications and services. An enterprise SSO can also be used to generate random passwords on their behalf. In the latter case, the user never knows the passwords to the individual applications and services. This can provide increased security, because the only way a user can gain access is via the enterprise SSO system, thereby preventing back-door access.

Facilities are also available to accommodate an enterprise that wants all the passwords for a user's applications and services to be the same. This approach, however, is not recommended because it can undermine security by providing a means of access that circumvents the enterprise SSO systems and the associated auditing and alert processes.

**Auditing and Alarms**

An enterprise’s SSO audit capability keeps track of all the actions carried out by users. Auditing capabilities ensure that security policies are enforced and that users are accountable for their activities. For example, the security policy may specify a maximum of three password attempts to log on to the network. When an enterprise SSO senses the fourth attempt, it will take whatever action has been stipulated. It could, for example, close the workstation down or raise an alert; an enterprise SSO can even raise a *silent* alert that allows the attacker to continue the break-in attempt unaware that he or she has been detected. This allows time to trace the offender.
The record of events can be analyzed to identify where breaches of security and policy may have occurred. For events where immediate attention is required, the audit facility can trigger alarms that can be directed to any number of places, including administrator workstations, a pager or e-mail system, or an event management system. Examples of information that can be audited include:

- end of a user session
- failed log-on attempts
- inactivity periods for workstations
- lock-outs of users and terminals
- services and applications used
- start of a user session and role adopted by the user
- system start-up and close-down
- use of administration utilities

A logged event can also specify the user, role, and workstation or terminal used, as well as the date, time, and session used. Audit administrators have considerable control over configuration of alarm capabilities and can set parameters on some 600 events to trigger the following actions:

- enter the event in an audit file
- ignore the event
- raise an alert

In addition to the deterrent effect of audit facilities, the audit log can be used to assess damage and recover data caused by breaches of security. It can also provide insight into the method of attack used, which can form the basis of effective future defense.

Furthermore, the security administration tools provide methods to access the audit file via various filters. In addition, the audit data can be moved into popular databases where ad hoc inquiries and various reports can be run.

**SSO Encryption**

All enterprise SSO components include encryption facilities that the system uses to enhance security. Techniques used include one-way encryption algorithms for password protection and two-way algorithms for functions such as password access control (PAC) and script-variable protection.
Integration of Other Authentication Techniques

An enterprise SSO itself offers standard authentication via password. It can also support additional or alternative authentication techniques, such as magnetic stripe cards and smart tokens (Security Dynamics SecurID card). If required, these may be associated with specific roles. In addition to an enterprise SSO password authentication method, it also provides APIs to other authentication facilities to enable an enterprise to integrate its own choice of authentication method.

There are other single sign-on solutions. In particular, consider next Sylogic B. V.'s single sign on product SSO+.

OTHER SSO SOLUTIONS

With the proliferation of client/server technologies over the past years, information has gradually migrated from centralized computers to heterogeneous, distributed environments. This trend toward open and diverse systems has impacted today’s enterprises by requiring that end users remember a number of IDs and passwords to gain access to various computing platforms and applications. The necessity for multiple passwords, while designed to increase security has instead compromised enterprise productivity and security, as well as increased the costs of managing a diverse computing environment.

How can multiple passwords compromise both productivity and security? The amount of time it takes for a user to recall and enter multiple passwords to access various platforms and applications can translate to tens of thousands of dollars a year. Users who desire to be more efficient by writing down their numerous passwords for swift recall can create a significant security breach. Add to both these scenarios the fact that today's help-desk administrators routinely claim that up to 50 percent of their time is spent resetting forgotten passwords. It all translates into a waste of time and productivity in a system that is far less secure than anyone can imagine.

Sylogic B.V.'s SSO+ is one of several SSOs described in this article that also addresses these concerns. As previously discussed, SSO reduces the frustration of end users by enabling them to easily access multiple platforms and applications through the use of a single password. It eases the burden of password administration for the help desk or system administrators by consolidating user account administration. It also offers enterprises the assurance that access across distributed computing environments remains secure.

For example, SSO+ is a product developed by IBM in The Netherlands and at Sylogic. It was designed to offer single log-on, additional security, and central user management, including authorizations in existing heterogeneous client/server environments, without having to make changes to the existing applications.
Single log-on is achieved using pass tickets. It allows the existing authentication methods of the server to continue to be used. This offers the advantage that during the implementation of SSO, no modifications need to be made to the existing security repositories. This way, a step-by-step implementation at the user or application level is provided.

Standards and standard components have been used as much as possible in constructing SSO+. The use of SSO+ has a number of advantages for both users and IT managers. Users only have to log in once and can subsequently start any application for which they are authorized.

Choosing the Right Single Sign-on Solution

The truth is, there are not many single sign-on solutions designed to fully adapt to the number of diverse operating systems and authentication methods available today. SSOs should be capable of handling a diverse security infrastructure, and provide a seamless interface across platforms and applications to allow administrators to learn one system, rather than many. The following are key components that should be part of any well-designed single sign-on solution:

- open architecture
- open authentication
- support for multiple log-in methods, including one-time passwords
- credentials forwarding
- support for multiple servers, clients, and hosts
- seamless user and administrative interface
- central administration

Open Architecture. An SSO solution designed for today and tomorrow is based on open standards. Open architecture ensures that the solution can be easily extended as an enterprise's security policy evolves. As standards change, an open architecture adapts with them.

Open Authentication. Support for a diverse range of authentication methods is critical. Popular platforms must be supported, such as NetWare, UNIX, and Windows NT. Broker-based authentication such as digital certificates must be supported, along with hardware (smart card) and software token-based methods. And — perhaps even more critical — such support must be easy to weave into an existing security infrastructure, as many enterprises are planning to move to one or more of these systems in the near future.

Support for Multiple Log-in Methods. With all of the diverse systems in use today, SSO must support log-in methods from passwords to tokens. Due to the all-too-common use of sniffers to detect pass-
words sent in plain text over the Internet and intranet, support for one-time passwords (OTPs) is an important means of providing enhanced system security.

**Credentials Forwarding.** In the past, SSO solutions relied on scripts to forward passwords to various and diverse applications and platforms. Today's solutions use Application Programming Interfaces (APIs) and easy-to-use Login Dialogs instead. A well-designed solution must allow these forwarding methods to work in parallel to suit the needs of a heterogeneous computing environment.

**Support for Multiple Servers, Clients, and Hosts.** Today's most popular platforms, including UNIX, Windows (3.x, 95, 98, 2000, and NT), and MVS, must be supported. Legacy systems must be supported as well.

**Seamless User and Administrative Interface.** A Windows 95/NT, 98/NT, or 2000/NT look-and-feel eases the learning curve for administrators and end users. The SSO solution should maintain a consistent look-and-feel across platforms. It must appear integrated with the operating system and transparent to the end user.

**Central Administration.** Finally, security or system administrators must be able to manage the SSO solution from a central location. The interface, again, must be seamless with the operating system, enabling the administrator to easily learn the process and become proficient quickly and efficiently.

**CONCLUSION AND SUMMARY**

In today's heterogeneous computing environments, end users frequently need access to applications and network resources running on multiple platforms and systems to perform their day-to-day responsibilities. As a result, end users must use multiple sign-on routines, userids, and passwords. This cumbersome management problem impedes productivity and compromises security when end users resort to writing down their passwords in an effort to keep track of them.

With that in mind, this article is directed toward enterprise executives, IT managers, security administrators, and others who want to know more about how different types of SSOs are designed to be simple to administer in even the largest, most complex networks. Different types of SSO tools are presented that allow effective administration of:

- applications
- menus
- roles
SSOs should provide a highly flexible set of rules that enable a clear definition of users' access authority, as well as the construction of roles that encompass numerous capabilities. Relationships between the different administrative functions should be a vital element in an SSO's ability to define which services can be accessed by individual users. The system should embody these relationships in the following ways:

- An application definition can refer to SAP variables (SVs).
- Any role can be restricted to a single application or initial menu.
- Applications and menus are allocated to these roles.
- Menus contain lists of applications and other menus.
- SAP variable values can be assigned to users, roles, or specific applications.
- Users are allocated roles according to their needs.
- Workstations identifiable by an SSO server can be permitted or denied access by specific roles; users can only use such workstations to run roles permitted on that workstation.

Also presented in this article is a high-level overview of SSO technology requirements to meet the diverse needs of the end user, the helpdesk/security administrator, and the security officer who sets overall enterprise security policy. The enterprise SSO should be able to meet these needs by increasing user productivity, reducing administrative overhead, and increasing overall security throughout the enterprise. Its flexible, open architecture should ensure that it is designed to meet the requirements of today's ever-evolving computing environment.

SSOs should also be able to automate user log-in to applications and platforms by supplying an application launch pad that acts as a familiar desktop for users and simplifies log-in, reducing it to a simple point-and-click process. End users authenticate once, and are presented with a customized desktop of authorized applications, that they can access quickly and efficiently. This simple process should enable an enterprise to flexibly move from password-based log-ins to strong authentication methods without visibly impacting the log-in process for the end user.

SSOs should be able to consolidate security administration by providing a centralized database of userids, application dialogs, access paths, and preferred credentials-forwarding information. An additional tier of administration enables central management of user access to applications and platforms across the enterprise.

An SSO should also be able to act as a mediator between users and applications, matching the best-available security with application capabilities.
and requirements. Most of the SSOs discussed in this article offer a hybrid approach to support SSO for a wide range of applications and systems, from legacy applications requiring proprietary passwords to one-time passwords to encrypted tickets. Finally, their architecture should be able to meet the requirements of the most complicated computing environments by providing a secure method of single sign-on that includes:

- a flexible use of current credentials-forwarding methods and adaptability to future security standards
- the ability for a phased implementation approach to ease the introduction of enhanced security standards across an enterprise
- an effective and secure response that meets the SSO expectations of end users, administrators, and security officers alike

John Vacca is an information technology consultant and internationally known author based in Pomeroy, Ohio. Since 1982, John has authored 28 books and more than 350 articles in the areas of Internet and Intranet security, programming, systems development, rapid application development, multimedia, and the Internet. John was also a configuration management specialist, computer specialist, and the computer security official for the NASA space station program (Freedom) and the International Space Station Program, from 1988 until his early retirement from NASA in 1995. John can be reached at jvacca@hti.net.

NOTES
2. This means that SSOs shall not require that all sign-on operations be performed at the same time as the primary sign-on operation. This would result in the creation of user sessions with all possible services although those services may not actually be required by the user.
3. Some authentication technology (e.g., those based on challenge-response mechanisms of which a user-held device is a component) may not be appropriate for use as part of secondary sign-on functions.
4. The Schumann answer for the dilemma of multiple sign-ons is Secure Single Sign-On (SAM/SSSO). This fully scalable solution uses a tamper-proof smart card (other tokens are possible) and a single sign-on PIN to access any environment from client/server to legacy. All log-ins and passwords are encrypted into the card's integrated circuit. Only possession of the card and knowledge of the single sign-on PIN will allow it to function. The smart card operates by presenting all log-ins and passwords immediately and automatically to applications in use, creating large gains in productivity and security.

Addresses of Entities