A SYSTEM FOR IDENTITY MANAGEMENT AND FORTIFICATION OF AUTHENTICATION

FIELD OF THE INVENTION

[0001] The present invention pertains to the field of secure networks and computing devices. More particularly, the present invention relates to automatic user authentication.

BACKGROUND OF THE INVENTION

[0002] With rapid growth of Internet and networks, the popularity of Internet technology rises among users of network services. In order to provide secure access to network services, user names and passwords are utilized to authenticate the user logging into a system providing particular network services. Users may accesses several applications, each with its own separate authentication mechanism causing the user to remember multiple user names and passwords. Due to this inconvenience users usually utilize the same user name and password for multiple applications that they access. In addition, users choose easy to remember passwords, which usually are easy to crack by hackers. Cracking of one password for an account breaches other accounts with the same user name and password. Network setups such as wireless Local Area Networks, remote access features, and weak intrusion protection increase vulnerability of passwords to technical attacks by hackers.

[0003] Many hackers are able to trick users by posing as system administrators causing the users to voluntarily provide the hackers with their passwords and user names.

[0004] Due to multiple accounts and multiple passwords that users maintain, password management for system administrators becomes a tedious and sometimes

of a departing employee are examples of tasks that system administrators need to perform in order to manage passwords of existing accounts in the system. Inaccurate password management may lead to security breaches, such as failing to delete a password of a fired employee may allow that employee to access network areas that that employee should not be accessing anymore.

[0005] Further, even if passwords are correctly managed, using passwords correctly for authenticating users is fundamentally vulnerable to various attacks from anywhere on the Internet. One of the best ways to lower the population of potential attackers is to use a certificate-based authentication mechanism with private keys stored on physical tokens. The process of transitioning from password-based authentication to token/certificate-based authentication is a complex process. However, it is a transition process that all enterprises serious about digital security need to undertake.

[0006] What is needed, therefore, is a solution that overcomes these and other shortcomings of the prior art.

SUMMARY OF THE INVENTION

[0007] A method and apparatus for automatic user authentication are described. The method may include collecting authentication credentials by monitoring authentication procedures of a plurality of applications accessed by a user and replacing the collected authentication credentials with stronger forms of credentials. The method may also include automatically utilizing the stronger forms of credentials to provide the user with access to the plurality of applications.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements.

[0009] Figure 1 illustrates an exemplary system architecture according to one embodiment of the invention;

[0010] Figure 2 illustrates components of an Access Agent according to one embodiment of the invention;

[0011] Figure 3 illustrates components of a Secure Object for Convenient Identification according to one embodiment of the invention;

[0012] Figure 4 illustrates components of Identity Management System according to one embodiment of the invention;

[0013] Figure 5 is a flow chart of a startup procedure according to one embodiment of the invention; and

[0014] Figure 6 is an exemplary architecture of a processing system according to one embodiment of the invention.

DETAILED DESCRIPTION

[0015] A method and apparatus for user authentication is described. Note that in this description, references to "one embodiment" or "an embodiment" mean that the feature being referred to is included in at least one embodiment of the present invention. Further, separate references to "one embodiment" in this description do not necessarily refer to the same embodiment; however, neither are such embodiments mutually exclusive, unless so stated and except as will be readily apparent to those skilled in the art. Thus, the present invention can include any variety of combinations and/or integrations of the embodiments described herein.

[0016] The present invention discloses a method and system for authenticating user via physicalization of user credentials. Passwords and usernames of a user are stored in a device and automatically provided to corresponding applications that the user is attempting to access.

[0017] It will be appreciated that the term "playback", as used herein, means automatically inserting stored user authentication information into appropriate applications. The term "client machine", as used herein, means a processing system hosting a Secure Object for Convenient Identification.

Related Technology

[0018] Introduction to related technology may be helpful in understanding some embodiments of the invention.

[0019] One embodiment of the invention utilizes Simple Object Access Protocol (SOAP). SOAP is a message-based protocol based on Extensible Markup Language

(XML) for accessing services on the Web. SOAP employs XML syntax to send text commands using HTTP.

[0020] One embodiment of the invention utilizes HyperText Transfer Protocol Secure (HTTPS). HTTPS is a protocol for accessing secure Web servers. Using HTTPS in a Uniform Resource Locator (URL) instead of HTTP directs the message to a secure port number rather than to a default port number.

Exemplary Architecture

[0021] Figure 1 illustrates an exemplary architecture of the invention. An Access Agent 100 communicates with Identity Management System (IMS) 110 via SOAP or HTTPS. IMS is located on a server machine. In addition, the Access Agent 100 interfaces with Secure Object for Convenient Identification (SOCI) device 120 via SOCI Application Program Interface functions. Figure 2 illustrates components of the Access Agent 100. In one embodiment the Access Agent 200 includes an installer 205, which installs the Access Agent 200 on a client machine hosting the SOCI. The Access Agent 200 includes a user interface module 210, which provides the end user with a graphical interface allowing management of the Access Agent's functions. The Access Agent 200 also includes a duplication module 215 that allows the user to perform duplication of the SOCI, description of which will be apparent from the following discussion. The Access Agent 200 may comprise a scripting tool module 220, which provides the end users with a mechanism to write new scripts to be utilized by the Access Agent 200 for managing passwords for new applications. A sniffer module 225 may also be included in the Access Agent 200 to capture user behavior and play back user authentication information. The Access Agent 200 also includes a session management module 230 to replace graphical

authentication interface in the system and provide session management control on the client machine. An Access Agent controller (AA controller) 235 ensures a proper startup of the Access Agent 200 upon an insertion of SOCI into the client machine. The Access Agent 200 also includes a data management module 240. The data management module 240 includes Certificate Management Module 260, Access Info Management Module 265, Configuration Management Module 270 and Audit Log Module 275. Certificate Management Module 260 manages data related to digital certificates such as parsing the certificate and generating a certificate request. The Access Info Management Module 265 manages data related to application access such as extracting user identification and password information. The Configuration Management Module 270 manages data related to configurable parameters of Access Agent. The Audit Log Module 275 manages logging of activities of the Access Agent for audit purposes. The Access Agent 200 also includes a synchronization module 245, communication module 250 and SOCI management module 255, functions of which will also be apparent from the following discussion.

[0022] Figure 3 illustrates an exemplary architecture of the SOCI according to an embodiment of the invention. The SOCI is a hardware token capable of being connected to the user's computer. The SOCI includes a chip 300. The chip 300 may be a smart card chip. The chip 300 includes a crypto processor 310 that performs cryptographic calculation described below. Cryptographic calculations include symmetric key, asymmetric key and hash algorithms such as RSA, DES, 3DES, SHA1 and MD5, all of which are well known in the art and do not require any further explanation. In addition, the chip 300 includes NVRAM to store sensitive private data, such as private keys. The

SOCI also includes Flash RAM 315 to store software drivers and non-sensitive data such as user configuration data, digital certificates, etc. The USB Flash controller 320 is another component of the SOCI. The USB Flash controller provides access from the client computer, i.e. SOCI host computer, to the Flash RAM storage 315 and the chip 300. The SOCI includes Application Interface Functions via which the client computer communicates with the SOCI. The Application Interface Functions provide high-level abstraction for SOCI services, such as certificate management, data encryption/decryption, and digital signature generation. The functions exposed by the Application Programming Interface may be implemented by a SOCI Runtime Library (not shown). In one embodiment, the SOCI stores its authentication information to be provided to the Access Agent in a certificate signed by Certificate Authority (CA) trusted by the Access Agent. The Certification Authority (CA) is an entity entrusted to issue certificates asserting that the recipient individual, machine or organization requesting the certificate fulfills the conditions of an established policy. Certificates together with private keys may be utilized in SOCI to authenticate the user.

[0023] Figure 4 illustrates the Identity Management System (IMS) 400 that is located on a server machine and communicates with the client machine that hosts the SOCI according to one embodiment of the invention. The Identity Management System includes registration services module 410, synchronization services module 415, administration services module 420 and backend services module 425. In addition, the IMS 400 includes a system configuration interface 426 allowing system operators to access administration services and backend services. In one embodiment the interface is an HTML interface. The registration services module 410 includes account registration

module 430, which registers a particular SOCI with the IMS. Once the SOCI is registered, the IMS may start provide services to the user of the SOCI. In addition, the account registration module 430 performs enterprise identity registration and email address registration. During the enterprise identity registration, the Access Agent provides IMS with existing credentials of an enterprise identity such as user identification and password information. IMS confirms the information by contacting the enterprise server and upon confirmation, IMS issues a certificate and registers the user in IMS database. During the email address registration, Access Agent provides IMS with an email address, upon receipt of which, the IMS sends an email message using the provided email address requesting identity verification. Once identity verification is received by the IMS, the IMS issues a certificate and registers the user in the IMS database.

[0024] The synchronization services module 415 includes synchronization module 435, update module 440 and escrow module 445. The synchronization module 435 synchronizes data on SOCIs with a copy on the IMS in order to maintain consistency between the data. The update module 440 updates software and data on the SOCI by downloading it to the SOCIs via the client machine hosting the SOCIs. For example, the update module 440 may download software upgrades, access scripts for common applications, administrator-maintained client configuration, etc. The escrow module 445 stores sensitive information on behalf of users for future recovery. The escrow module 445 stores the Common Symmetric Key (CSK) in an encrypted form. In a situation where all SOCIs are lost and data is not recoverable from any of these devices, the encrypted data stored in IMS can be recovered by restoring CSK from the escrow module 445. The escrow module 445 may keep the CSK of the SOCI in an encrypted form that is

recoverable when IMS's private key is provided to the escrow module 445 and the user presents a predetermined password phrase. CSK is an encryption key used to encrypt user's authentication information such as user's passwords. Every SOCI device that belongs to the same user has the same CSK so that information encrypted by on SOCI can be decrypted by another SOCI belonging to the same user.

[0025] The backend services module 425 includes certificate management module 450 and data module 455. Certificate management module 450 issues new certificates, revokes certificates and maintains Certificate Revocation Lists (CRLs) for certificate validity verification. The data module 455 provides other IMS modules with access to the data stored in IMS databases. IMS database may reside in Relations Database Management System (RDBMS), directory server or a simple file system. The databases contain information about SOCI devices that the user owns, such as serial number and issued certificate. The databases may also include information about the user, such as applications that the user accesses, encrypted passwords and configuration data.

[0026] The administration services module 420 includes user administration module 460. The user administration module 460 allows the administrator of the system to create new user accounts, delete existing accounts, assign roles to users, i.e. specify users with administration authorization, bind users to accounts, i.e. email accounts. In addition, the administration module 460 allows the administrators to configure SOCIs before distribution, create new key pairs, i.e. public and private keys, revoke existing certificates and keys. Generation of public and private key pairs may be performed in SOCIs. Public keys can then be stored in a certificate issued by IMS. Theses certificates, together with private keys stored in SOCI can be used to authenticate SOCI to IMS in

order to retrieve data. In addition, certificates along with private keys may also be used to authenticate user to applications that utilize certificate-based authentication mechanisms.

[0027] The physical processing platforms that embody the Access Agent and IMS may include processing systems such as conventional personal computers (PCs) and/or server-class computer systems according to various embodiments of the invention. Figure 6 illustrates an example of such a processing system at a high level. The processing system of Figure 6 includes one or more processors 600, read-only memory (ROM) 610, random access memory (RAM) 620, and a mass storage device 630 coupled to each other on a bus system 640. The bus system 640 includes one or more buses, which may be connected to each other through various bridges, controllers and/or adapters, which are well known in the art. For example, the bus system 640 may include a 'system bus', which may be connected through an adapter to one or more expansion, such as a peripheral component interconnect (PCI) bus or an extended industry standard architecture (EISA) bus. Also coupled to the bus system 640 are a the mass storage device 630, one or more input/output (I/O) devices 650 and one or more data communication devices 660 to communicate with remote processing systems via one or more communication links 665 and 670, respectively. The I/O devices 650 may include, for example, any one or more of a display device, a keyboard, a pointing device (e.g., mouse, touchpad, trackball), an audio speaker.

[0028] The processor(s) 600 may include one or more conventional general-purpose or special-purpose programmable microprocessors, digital signal processors (DSPs), application specific integrated circuits (ASICs), or programmable logic devices (PLD), or a combination of such devices. The mass storage device 630 may include any

one or more devices suitable for storing large volumes of data in a non-volatile manner, such as magnetic disk or tape, magneto-optical storage device, or any of various types of Digital Video Disk (DVD) or Compact Disk (CD) based storage or a combination of such devices.

[0029] The data communication device(s) 660 each may be any devices suitable for enabling the processing system to communicate data with a remote processing system over a data communication link, such as a wireless transceiver or a conventional telephone modem, a wireless modem, an Integrated Services Digital Network (ISDN) adapter, a Digital Subscriber Line (DSL) modem, a cable modem, a satellite transceiver, an Ethernet adapter, or the like.

Methodology

[0030] With theses concepts in mind embodiments of the invention can be further explored.

Startup Procedure

[0031] In order for a user to be automatically authenticated for each application that the user attempts to access, the Access Agent 200 can be executing on the user's machine, i.e. client machine. The startup procedure will be described with reference to Figure 5. At 510 the session management module 225 is executed upon the boot up of the client machine. The session management module 225 interacts with a logon procedure of the operating system to handle initialization procedures. The initialization procedures are the following. The session management module invokes the AA controller 235. Upon initialization, the AA controller 235 at 520 directs the session management module 230 to start a thread, which may poll USB ports of the client machine. The polling thread

identifies whether an SOCI is present in any of the USB ports. If the polling thread does not identify the SOCI, the session management module 230 at 525 prompts the user to insert the SOCI and awaits for the insertion of the SOCI by periodically polling the USB ports. If the polling thread identifies that SOCI is already connected to the USB port or if the new SOCI has been inserted, the session management module 230 displays a dialogue box prompting the user for a personal identification number (PIN). Upon the user entering the PIN, the session management module 230 at 535 invokes the SOCI management module 255 to verify the entered PIN. If the PIN is successfully verified the SOCI management module 255 provides the session management module 230 with the operating system login and password information of the user at 540. For example, if the client machine is running Windows Operating System, the SOCI management module 255 provides the session management module 230 with Windows Login ID and Windows Password. In one embodiment the operating system login identification and password data are encrypted and stored in the SOCI and retrieved by the SOCI management module 255 via SOCI APIs. The user may have several operating system login identifications and passwords and in this case the user may be presented with a pull down menu to select the login ID and password for the current session. At 545 upon determining and decrypting the login ID and password, the session management module 230 inserts the ID and password into the operating system logon procedure.

[0032] After successful logon, the session management module 230 invokes the user interface module 210, invokes the sniffer module 225 and the synchronization module 245.

SOCI Initialization

[0033] In one embodiment upon insertion of the SOCI, a setup program located in the flash memory of the SOCI is executed to determine whether the Access Agent 200 is installed on the client machine. If the Access Agent is not installed on the client machine, the setup program locates the download server to download the access agent installer module. The setup program may contain a default location of the installer module. If the setup program fails to locate the installer for download, the setup program prompts the user for location of the installer or for an insertion of a diskette or CD-ROM including the installer module. Upon installation of the installer, the user is prompted to enter an SOCI personal identification number (PIN) and password. PIN of the SOCI is distributed with the SOCI. User can change the PIN after obtaining access to the SOCI upon entering the original PIN. Upon the user entering the PIN and password, the installer transmits the PIN and password data to the IMS. In one embodiment data transmitted to the IMS includes SOCI identification number retrieved from the SOCI device, SOCI properties, SOCI public keys, encrypted Common Symmetric Key (CSK). Upon receiving the data, the IMS creates a new user account and registers the SOCI with the account. The IMS generates a new certificate and transmits the certificate to the Access Agent which stores the certificate in the SOCI. The IMS may also encrypt the CSK with a key derived from the SOCI password and further encrypt the CSK with the IMS's public key. In one embodiment, the server's public key is stored on a separate secure server, or stored in a hardware key device.

Automated Authentication

In one embodiment the sniffer module 225 of the Access Agent 200 executes in the background at the client machine and identifies user's login, logout, change of password

activities and records the procedures in a form of an access script. The access scripts are encrypted and stored in the SOCI and the IMS server. The sniffer module 225 captures operating system messages for various applications and identifies whether any of the captured messages comprise user authentication data. If the sniffer module 225 identifies the user authentication application data for a particular application, the sniffer module 225 stores the information in the SOCI. Upon identifying the user authentication application, the sniffer module 225 generates access scripts to be played back when the user attempts to access an application requiring authentication information. When the user attempts to access the application, the sniffer module 225 determines whether an access script exists for the application. If the access script exists, the sniffer module 225 injects the authentication information into the login procedure of the application. If the access script does not exist, the sniffer module 225 captures the logon information entered by the user and stores the encrypted information in the SOCI and IMS. An access script is an xml-based script that contains information on how to playback authentication information, such as the location of the application in the computer, the name of the application, the buttons to click, etc. An example of an access script is provided below:

<AccessScript ASPoint="explorer.exe">
<ASMethod MethodName="explorer.exe-1" MethodType="login"><ASStep ID="1"><ASResult>
<WebSignature><PageURL></PageURL>
<UserFieldName></UserFieldName>
<PwdFieldName>
</PwdFieldName>
<ActionFieldName></ActionFieldName></WebSignature>
<WndSignature><WndID/>
<WndTitle>Connect to</WndTitle><ServerLabel></ServerLabel><UserNameLabel>User name:</UserNameLabel><PasswordLabel>Password:</PasswordLabel><New PasswordLabel></NewPasswordLabel><VerifyPasswordLabel></VerifyPasswo rdLabel><LeftStr>Connect

to</LeftStr><RightStr></RightStr><ServerDlgID/><UserNameDlgID/><New PasswordDlgID/><OkButtonID/></WndSignature><ASEvent><Message></Message></ASEvent></ASResult></ASStep></ASMethod></AccessScript>

[0034] In addition, the access script contains information allowing the sniffer module to recognize access points of an application, the class identification of the application, password policies associated with the application, etc.

[0035] The sniffer module 225 may also perform a client-side single sign-on, by acting as a single sign on service upon the user inserting the SOCI and entering the PIN to unlock the information stored in the SOCI. The sniffer module 225 plays back credentials to a plurality of applications that the user accesses.

[0036] In one embodiment upon identification of user's authentication data, the sniffer module 225 converts the user's authentication data into a stronger form of authentication data to be then presented to the applications that user attempts to access. The conversion of the authentication data may be performed without the user being aware the change. The sniffer module 225 can generate a longer password by adding alpha-numeric characters into the password, for example to the end of the user's password. The sniffer module 225 can also generate a random password to be utilized for user authentication purposes instead of the user's chosen password to ensure higher security levels. The new password is generated base on configurable criteria such as the minimal length, or the inclusion of special characters. In addition, the stronger form of authentication data can be digital certificates, private keys, etc. The request for change of passwords to the application can be performed by either Access Agent or IMS. This is done by supplying both the old password and the new password to the application. Once the application accepts the change and is aware of the new password, Access Agent will

store the new password in the form of configuration data encrypted by the CSK. The sniffer module 225 may also request IMS for a digital certificate using a private key stored in the SOCI, This stronger form can be used for user authentication purposes instead of user's password if the application is converted to used public key authentication mechanism. Once again, the procedure of conversion of user's password into a stronger form of authentication credentials may be performed without knowledge of the user. By configuring the Access Agent to periodically and automatically perform the above procedures, user credentials will be more secured, hence they are fortified.

Data Synchronization

[0037] In one embodiment the user authentication data and access scripts are stored on SOCI and on the IMS server for a backup. The data on the SOCI and IMS server is identical, unless during one of the update sessions by the sniffer module 225, the server was not accessible due for example, to lack of network connection between the client machine and the IMS server. Also, the data on the server may be updated when the user utilizes a duplicate SOCI, causing the original SOCI not to have the latest copy of the user authentication data. In one embodiment, all the records stored in the SOCI and IMS server are time stamped allowing the synchronization module 245 to determine whether SOCI or IMS server includes the latest data. Upon determining the location of the latest user authentication data, the synchronization module 245 updates the data to ensure identical copies of user authentication data on SOCI and IMS server.

[0038] In one embodiment, the user authentication data may be stored on the client machine as software. If an SOCI device is not available, the user may request the stored authentication data from the IMS server. Upon downloading the user

authentication information to the client machine, the downloaded data may be used by the Access Agent in a manner described above.

[0039] In addition, the data stored at the IMS server may be downloaded to a new SOCI acquired by the user. The information stored in the SOCI to be replaced by the new one is encrypted and uploaded to a server, which may be the IMS server. The original SOCI exports the CSK encrypted using the new SOCI's public key. The new SOCI downloads the encrypted CKS. Once the encryption key is acquired by the new SOCI, the encrypted authentication data is downloaded from IMS to the new SOCI to be decrypted utilizing the encryption key. The new SOCI is therefore able to access the same information as the original SOCI, and is said to host a cloned credential container.

Public/Private Key Authentication

[0040] In one embodiment of the invention, SOCIs include public-private key pairs to be registered with a Certificate Authority of IMS. The issued certificate and key pair are stored in the SOCI. When the Access Agent detects an application that has been configured to employ public keys for user authentication, the Access Agent directs the SOCI to perform crypto function to automatically cause the application to provide the user with the access. The private key is stored in the SOCI and is not provided to any application or any user. The SOCI has physical tamper-proof features to ensure that private keys are not released. In one embodiment the private key may be burned into the chip of SOCIs during manufacturing.

[0041] In one embodiment administrators of IMS may cause the authentication system to utilize private-public key method without the system users being aware of the change. Due to automatic user authentication, the users need not be aware of the

authentication method employed as long as they are provided with the desired application access.

Manual Creation of SOCI Contents

[0042] In one embodiment the user authentication data is downloaded to SOCIs from a database manually created by system administrators. System administrators create user name and password data pairs for each user of the system and store the authentication data in a database that may be stored at a server or at a corresponding computer of each user. Upon the user connecting an SOCI to the user's computer, the authentication information from the database stored in the usr's computer is downloaded to the SOCI. Alternatively, system administrators download the created authentication data from the database stored at the server to corresponding SOCIs prior to distributing SOCIs to the users. and upload the created authentication data to a corresponding SOCI for each user.

Single Administrative View

[0043] In one embodiment, the synchronization module of the Access Agent uploads credentials of all applications that the user uses to the IMS. IMS organizes the uploaded information and presents information about each application to an administrator of user's system upon request. Therefore, a single consolidated user directory can be created that contains information across a plurality of applications. The administrator of IMS will use this consolidated directory; directories of each individual application will no longer be necessary. In one embodiment, the administrator can remove these directories and effectively consolidate them into the single user directory. The administrator is able to remove access to all applications that the user is accessing with the provided

information. This may be done automatically through an interface allowing removal of user access to applications. Alternatively, this can be done manually by the administrator.

[0044] In one embodiment, the administrator of IMS can provision access to a plurality of applications to the user. IMS can create accounts in applications and inject authentication information of the newly created account into the credential container store on the server. The data synchronization module of the Access Agent downloads the information and instructs the playback module of the Access Agent to utilize the downloaded information to access the newly created accounts.

Session Challenge/Response

[0045] In one embodiment, the sniffer module 225 of the Access Agent detects that a web page that a user is attempting to access contains embedded XML tags indicating that the application requires strong authentication through Session Challenge response. The sniffer module 225 contacts the application server to present a certificate. The application issues a challenge to the Access Agent, requiring the Access Agent to digitally sign a random datum with the private key. The Access Agent signs the datum using the information stored in SOCI. The applications returns a session identification to the sniffer module 225 allowing the user to access the application.

[0046] It will be appreciated that user authentication information does not have to be stored in a hardware token, such as SOCI, but maybe stored in a database located at a server. In addition, it will be appreciated that user authentication information does not need to be converted into a stronger form of authentication and original user authentication information can be played back by the sniffer module 225.

[0047] Thus, a method and apparatus for user authentication have been described. Although the present invention has been described with reference to specific exemplary embodiments, it will be evident that various modifications and changes may be made to these embodiments without departing from the broader spirit and scope of the invention as set forth in the claims. Accordingly, the specification and drawings are to be regarded in an illustrative sense rather than a restrictive sense.

CLAIMS

What is claimed is:

1. A method comprising:

collecting authentication credentials by monitoring authentication procedures of a plurality of applications accessed by a user;

replacing the collected authentication credentials with stronger forms of credentials; and

automatically utilizing the stronger forms of credentials to provide the user with access to the plurality of applications.

2. The method of claim 1 wherein the collecting of the authentication credentials includes monitoring user's procedures of changing authentication credentials.

3. The method of claim 1 wherein the collecting the authentication credentials includes collecting authentication credentials manually inputted by the user.

4. The method of claim 1 wherein the collecting the authentications credentials includes collecting authentication credentials inputted by a system administrator.

5. The method of claim 1 wherein the replacing the collected authentication credentials is performed without knowledge of the user.

6. The method of claim 1 wherein the authentication credentials include user name and password pairs.

7. The method of claim 1 wherein the authentication credentials include private keys.

8. The method of claim 1 wherein the authentication credentials include digital certificates.

9. The method of claim 1 wherein the stronger forms of credentials include longer passwords.

10. The method of claim 1 wherein the stronger forms of credentials include random passwords.

11. The method of claim 1 wherein the stronger forms of credentials include private keys and digital certificates.

12. The method of claim 1 further comprising storing the authentication credentials in a hardware token.

13. The method of claim 1 further comprising storing the authentication credentials in a data storage.

14. A method comprising:

monitoring procedures of changing authentication information performed by a user;

automatically generating stronger forms of the new authentication information;

replacing old authentication information stored in a credential data storage with the stronger forms of the new authentication information.

15. The method of claim 14 wherein the authentication information includes user name and password data pairs.

16. The method of claim 14 wherein the authentication information includes private keys.

17. The method of claim 14 wherein the authentication information includes digital certificates.

18. The method of claim 14 wherein the credential data storage is a hardware token.

19. The method of claim 14 wherein the credential data storage is a database.

20. The method of claim 14 wherein the authentication information includes authentication credentials utilized to access an application.

21. A method comprising:

uploading a first credential data storage including encrypted user authentication information;

transferring an encryption key to a second credential data storage;

downloading the encrypted authentication information to the second credential storage.

22. The method of claim 21 further comprising uploading the first credential data storage to a server.

23. The method of claim 21 further comprising downloading the encrypted authentication information from the server.

24. The method of claim 21 wherein the credential data storage is a hardware token.

25. The method of claim 21 wherein the credential data storage is a database.

26. The method of claim 21 wherein the authentication data is a user name/password data pair.

27. The method of claim 21 wherein the authentication data is a private key.

28. The method of claim 21 wherein the authentication data is a digital certificate.

29. A method comprising:

collecting authentication credentials for a plurality of applications accessed by a user;

storing the authentication credentials in a credential container;

playing back the authentication credentials for the plurality of applications upon subsequent user accesses.

30. The method of claim 29 wherein the credential container is created by an administrator.

31. The method of claim 29 wherein the credential container is distributed to the user by the administrator.

32. The method of claim 29 wherein the credential container is a hardware token.

33. The method of claim 29 wherein the credential container is a database.

34. The method of claim 29 wherein the playing back the authentication credentials includes inserting authentication credentials corresponding to an application to be accessed by the user.

35. The method of claim 29 wherein collecting the authentication credentials includes monitoring authentication procedures of the plurality of applications accessed by the user.

36. A method comprising:

storing authentication credentials for a plurality of applications accessed by a user;

organizing the contents of the authentication credentials to provide a view of all applications accessed by the user;

removing access privileges to at least one application of the plurality of applications.

37. The method of claim 36 wherein authentication credentials are stored on a server.

38. The method of claim 36 wherein the view of all applications accessed by the user is presented to an administrator.

39. The method of claim 36 wherein removing access privileges to the at least one application is performed automatically.

40. The method of claim 36 wherein removing access privileges to the at least one application is performed manually by an administrator.

41. The method of claim 36 further comprising providing a consolidated view of applications accessed by the user, wherein user directories of individual applications are removed.

42. The apparatus of claim 40 wherein monitoring module further configured to monitor procedures of changing authentication credentials.

43. The apparatus of claim 40 the replacement module further configured to replace the credentials without knowledge of the user.

44. The apparatus of claim 40 wherein the authentication credentials include username and password pairs.

45. The apparatus of claim 40 wherein the stronger forms of credentials include longer passwords.

46. The apparatus of claim 40 wherein the stronger forms of credentials include private keys and digital certificates.

47. The apparatus of claim 40 wherein the monitoring module further configured to store the authentication credentials in a hardware token.