SYSTEM AND METHOD FOR INTEGRATED SECURITY ROLES

BACKGROUND OF THE INVENTION

Technical Field

The present invention relates in general to a system and method for integrating security roles across application boundaries. More particularly, the present invention relates to a system and method for expanding a security role by mapping an upstream security role to a downstream security role.

Description of the Related Art

Computer systems use various techniques for user authorization. A computer system typically authenticates and authorizes a user when the computer system receives a request from the user. For example, a user may access his bank account information and a banking application proceeds through a series of steps to authenticate and authorize the user, such as requesting a user identifier and a password from the user. The computer system may also require authorization to access downstream applications. Using the example described above, the banking application (i.e. upstream application) may call a downstream application to retrieve account information corresponding to the user's request whereby the downstream application requires user identification information in order to grant access.

Java 2 Enterprise Edition (J2EE) includes a security role-based access control mechanism for user request authorizations. A security role may be viewed as a collection of Enterprise Java Bean (EJB) method permissions along with read/write access permissions to URL pages. EJB beans and URL pages are packaged together into a J2EE application in order to build a functionality set that solves a business problem. System administrators map individual user identifiers as well as group identifiers to each security role in order to provide each user and group with required permissions to access business functions.

A challenge found with mapping users to applications, however, is that different developers or vendors create separate J2EE applications that are integrated into a larger business application. Security roles are typically defined within an application's boundary that, in turn, allows business applications to be developed in a modular fashion. For example, a business operation may be partitioned into a set of components

and different groups may develop the components separately. A group that develops a particular module may not be aware of users that require access from other modules.

Furthermore, a challenge found with manually managing user-to-role mapping in a modular system for downstream applications is that it becomes non-trivial and complex. For example, a workflow user request may be handled by a number of J2EE applications. When this occurs, the user's identity is mapped to security roles corresponding to each application in order to provide the user with access.

What is needed, therefore, is a system and method that automatically maps an upstream application's security roles to a downstream application's required security role and, conversely, maps downstream application security roles to upstream security roles. Moreover, what is needed is a system and method to more effectively provide users with access across application boundaries.

SUMMARY

Accordingly, according to a first aspect the present invention provides a method of authorizing a client's request at a first downstream application, said method comprising: receiving, at the first downstream application, a first application request from an upstream application, wherein the first application request is derived from the client's request and includes an upstream security role identifier that was determined by the upstream application; reading authorization table entries stored in a downstream authorization table accessible from the downstream application; matching the upstream security role identifier included in the request with at least one of the authorization table entries included in the downstream authorization table; and authorizing the client's request in response to the matching.

According to a second aspect the present invention provides an information handling system comprising: one or more processors; a memory accessible by the processors; one or more nonvolatile storage devices accessible by the processors; and a client authorization tool to authorize a client's request at a first downstream application, the client authorization tool including: means for receiving, at the first downstream application, a first application request from an upstream application, wherein the first application request is derived from the client's request and includes an upstream security role identifier that was determined by the upstream application; means for reading authorization table entries

stored in a downstream authorization table accessible from the downstream application, the downstream authorization table located in one of the nonvolatile storage devices; means for matching the upstream security role identifier included in the request with at least one of the authorization table entries included in the downstream authorization table; and means for authorizing the client's request in response to the matching.

According to a third aspect the present invention provides a computer program product stored on a computer operable media for authorizing a client's request at a first downstream application, said computer program product comprising means for carrying out a method according to the first aspect.

Preferably one or more required downstream security roles are identified wherein each authorization table entry corresponds to at least one of the required downstream security roles. Optionally the required downstream security role that corresponds to the matched authorization table entry is selected and included in a second application request which is sent to a second downstream application.

Alternatively the matched authorization table entry is generated, prior to receiving the first application request. To generate the matched authorization table a role-mapping requirement is received which includes an upstream security role and a downstream security role, the upstream security role corresponding to the upstream security role identifier, and then the upstream security role identifier is included in the matched authorization table entry, the matched authorization table entry corresponding to the downstream security role.

Optionally it is determined whether the upstream security role and the downstream security role are equivalent, and the upstream security role and the downstream security role are configured such that they are equivalent.

Preferably the first application request is received over a computer network.

According to a fourth aspect the present invention provides a method of expanding an upstream security role to include a downstream application, said method comprising: receiving a security role-mapping request that corresponds to an upstream application and the upstream security role; selecting a downstream security role corresponding to the security role-mapping request and the downstream application; and adding

one or more downstream authorization table entries to the downstream security role, wherein the authorization table entries correspond to the upstream security role.

Preferably at least one of the authorization table entries is selected from the group consisting of a user identifier, a group identifier, and an upstream security role identifier.

Preferably it is determined whether the upstream security role and the downstream security role are equivalent, and the downstream authorization table entries and one or more upstream authorization table entries are configured such that they are equivalent, the upstream authorization table entries corresponding to the upstream security role.

Optionally an application request which includes a user identifier is received from the upstream application. In this case the user identifier is matched with one or more of the downstream authorization table entries and the application request is authorized in response to the matching.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention will now be described, by way of example only, with reference to a preferred embodiment thereof, as illustrated in the accompanying drawings, in which:

Figure 1 is a diagram showing an upstream application sending an application request that includes upstream security roles to a downstream application;

Figure 2 is a diagram showing an upstream application sending a request to a downstream application wherein the downstream application's corresponding authorization table includes one or more upstream subjects;

Figure 3A is a diagram showing an upstream security role identifier included in a downstream authorization table entry during a role-to-role mapping process;

Figure 3B is a diagram showing an upstream security role being mapped to a downstream security role by adding upstream security role user-to-role mappings to the downstream security role during a role-to-role mapping process;

Figure 4 is a high level flow chart showing steps taken in preparing code and updating security roles during phases in an application's lifecycle;

Figure 5 is a flowchart showing steps taken in analyzing upstream applications and role-mapping upstream security roles to downstream security roles;

Figure 6 is a flowchart showing steps taken in role-mapping upstream security roles to downstream security roles;

Figure 7 is a flowchart showing steps taken in a downstream application authorizing a user with an upstream security role assignment; and

Figure 8 is a block diagram of an information handling system capable of implementing the present invention.

The use of the same reference numerals in different drawings indicates similar or idential items.

PREFERRED EMBODIMENT OF THE INVENTION

The following is intended to provide a detailed description of an example of the invention and should not be taken to be limiting of the invention itself. Rather, any number of variations may fall within the scope of the invention which is defined in the claims following the description.

Figure 1 is a diagram showing upstream application **120** sending application request **130** which includes upstream security role **137** to downstream application **140**. The downstream application's corresponding downstream authorization table **170** has been role-mapped with an upstream security role identifier which corresponds to the upstream security role. The example in **Figure 1** shows that downstream authorization table **170** includes an upstream security role identifier in downstream authorization table entry **190** (see **Figures 3A**, **4**, **5**, **6**, and corresponding text for further details regarding role-mapping additions).

Client **100** sends request **105** to server **115** through computer network **110**, such as the Internet. For example, client **100** may wish to check a bank account balance which is accessed using server **115**. Request **105** includes user data **107** that server **115** uses to authenticate and authorize

client **100**. For example, user data **107** may include a user identifier, a password, a digital certificate, or other information used to authenticate and authorize the client.

Server **115** includes upstream application **120** which interfaces with client requests. Using the example described above, upstream application **120** may be an online banking application which allows users to check bank account balances, deposit funds, and withdraw funds. Upstream application **120** includes web module **122** and EJB (Enterprise Java Bean) module **125**. Web module **122** provides a user interface to client **100** (e.g. web page) and also authenticates client **100** using client **100**'s authentication data included in request **105** (e.g. a digital certificate). Once web module **122** authenticates client **100**, web module **122** calls EJB module **125** to further process request **105**.

Enterprise Java Bean (EJB) modules authorize a user's request based upon the user's assigned security roles. Using the example described above, EJB module **125** authorizes client **100** to check bank account balances if client **100** is assigned a "CheckBalance" role. In the example shown in **Figure 1**, EJB module **125** grants user authorization if client **100** is assigned the appropriate upstream security role **160** stored in data store **150**. EJB module **125** accesses upstream authorization table **155** located in authorization table store **150** during user authorization steps. Upstream authorization table **155** includes upstream security role **160** and corresponding upstream authorization entries **162**, **165**, and **167**. Upstream authorization table entries **162** and **165** include entries that are user identifiers whereas upstream authorization table entry **167** includes an entry that is a group identifier. Authorization table store **150** may be stored on a non-volatile storage area, such as a computer hard drive.

EJB module **125** determines whether to grant access to client **100** by comparing client **100**'s user identifier included in user data **107** with upstream authorization table entries **162**, **165**, and **167**.

Once EJB module **125** grants access to client **100**, EJB module **125** determines that it should call downstream application **140** to handle the request. Using the example described above, the online banking application determines that it should call a naming space management application in order to look-up user account information from a naming space. EJB module **125** sends request **130** to downstream application **140**. Request **130** includes user identifier **135** and roles **137**. User identifier **135** identifies client **100** and roles **137** includes identifiers corresponding

to client **100**'s upstream security roles, such as upstream security role **160**.

In one embodiment, downstream application **140** may be located on a separate server and roles **137** is included in a security context that is sent to the server hosting the downstream application over a computer network, such as a LAN or the Internet.

Downstream application **140** includes EJB module **145** which receives and analyzes request **130**. EJB module **145** determines that client **100** requires a particular security role assignment in order for EJB module **145** to grant access. Using the example described above, EJB module **145** grants access to look-up the user's bank account information from the naming space if the requestor is assigned downstream security role **175**. EJB module **145** accesses downstream authorization table **170** that includes downstream security role **175**. EJB module **145** compares user identifier **135** and roles **137** with downstream authorization table entries **180**, **185**, and **190**. Downstream authorization table entries **180** and **185** include downstream subjects that are user identifiers whereas downstream authorization table entry **190** includes an upstream security role identifier corresponding to upstream security role **160**. EJB module **145** determines that the upstream security role included in roles **137** matches the upstream security role identifier included in downstream authorization table entry **190** and grants access to client **100**.

In one embodiment, a computer system may map upstream subjects associated with a particular upstream security role to a downstream authorization table in lieu of adding an upstream security role identifier to the downstream authorization table (see **Figures 2**, **3B**, and corresponding text for further details regarding upstream user-to-role mappings to downstream security roles).

Figure 2 is a diagram showing an upstream application sending a request to a downstream application wherein the downstream application's corresponding authorization table includes one or more authorization table entries corresponding to the upstream application. **Figure 2** is the same as **Figure 1** with the exception that **Figure 2**'s downstream authorization table **270** was user-to-role mapped to an upstream security role (**260**) whereas **Figure 1**'s downstream authorization table **170** was role-mapped using an upstream security role identifier. In particular, downstream authorization table **270** includes authorization table entries **290**, **295**, and **299** that were added to downstream authorization table **270** during a

user-to-role mapping process (see **Figure 3B** and corresponding text for further details regarding user-to-role mappings).

Since downstream authorization table **270** was user-to-role mapped to upstream security role **260**, request **230** does not included client **200**'s assigned upstream security roles whereas **Figure 1**'s request **130** included client **100**'s assigned upstream security roles (e.g. roles **137**).

Figure 3A is a diagram showing an upstream security role identifier included in a downstream authorization table entry during a role-to-role mapping process. Upstream authorization table **300** includes upstream security role **305** that requires role-mapping to downstream security role **330**. One approach to interpret role-to-role mapping is that by role-mapping an upstream security role to a downstream security role, the downstream security role's permissions are included in the upstream security role. For example, if a user is assigned an upstream security role, the user is automatically granted access to a downstream security role that has been role-mapped to the upstream security role.

A server accesses upstream authorization table **300** which includes security role **305** and security role **315** during a role-to-role mapping process. Authorization table entries **310** include entries corresponding to security role **305** and authorization table entries **320** include entries corresponding to security role **315**.

The server accesses downstream authorization table **325** during role-to-role mapping and selects upstream security role **305** to map to downstream security role **330**. Downstream security role **330** includes authorization table entries **335** and **340**. The server adds authorization table entry **345** to downstream security role **330** in order to "role-map" upstream security role **305** to downstream security role **330**. Downstream authorization entry **345** includes an upstream security role identifier corresponding to upstream security role **305**. Downstream security role **330** is now configured so a downstream application grants access to users that are assigned upstream security role **305** (see **Figures 1, 7,** and corresponding text for further details regarding downstream application access granting steps).

Figure 3B is a diagram showing upstream security role **355** being mapped to downstream security role **375** by adding upstream security role user-to-role mappings (i.e. user identifiers and group identifiers) to the downstream security role during a role-to-role mapping process.

A server accesses upstream authorization table **350** which includes security role **355** and security role **365** during a role-to-role mapping process. Authorization table entries **360** include a collection of user-to-role mappings corresponding to security role **355**. Each entry in a user-to-role mapping table includes a user, a group, or a security role identifier that has been assigned to upstream security role **355**. The server accesses downstream authorization table **370** and selects downstream security role **375** to role-map upstream security role **355**.

Downstream security role **375** includes authorization table entries **385** and **390**. Authorization table entries **395** include new entries resulting from upstream security role **355**'s user-to-role mappings to downstream security role **375**. Downstream security role **375** is now configured so the corresponding downstream application grants authorization to users that are assigned upstream security role **355** (see **Figures 2**, **7**, and corresponding text for further details regarding downstream application authorization).

Figure 4 is a high level flow chart showing steps taken in preparing code and updating security roles during phases in an application's lifecycle. An application's life cycle includes an application development phase (e.g. design and coding), an assembly phase (e.g. packaging and defining policy and reference), a deployment phase (e.g. installation and filling in binding information), and a runtime phase (i.e. execution). The phases described above are J2EE terminology and as one skilled in the art can appreciate, other software development systems may be used to implement this invention.

An application developer is aware that his application (i.e. upstream application) requires access service from other applications (i.e. downstream applications). Therefore, the application developer defines security role references to the downstream applications that his upstream application requires access (step **400**). The application developer adds a security-role-reference extension to his code which defines a symbolic role name that may be used to reference security roles in downstream applications.

Since actual downstream security role names may not be known during an upstream application's assembly time, a security-role-reference element provides an indication that an upstream security role should be mapped to a downstream security role. Actual downstream security role names may be resolved during the upstream application's deployment phase (see below for further details). A "security-role-reference" may be defined as an

extension in a J2EE deployment descriptor. The "security-role-reference" may be added to EJB modules and Web modules, and defined as follows:

<!ELEMENT security-role-ref (description?, security-role-ref-name)>

The element may include an optional "description" to describe the downstream application to provide role-to-role mapping guidance to an application deployer during a binding process (see below for further details regarding binding process).

The application developer may add another extension to the J2EE deployment descriptor to allow a bind between a downstream security role and the security-role-reference. For example, an application is uniquely identified by a cell name and its application name. A cell includes a number of application servers and is a basic unit for setting up a security policy. A security role is uniquely identified by a three-tuple, such as:

(cell name, application name, security role name)

A deployer, when installing a new application, resolves security-role-references by adding an element in into a binding extension to the J2EE deployment descriptor as follows:

<!ELEMENTS security-role-binding (security-role-ref-name, unique-role-name)>

The above statement binds a unique role name (a three-tuple) with the security-role-reference name defined by the assembler (step **450**). An access control security policy that assigns users and user groups to security roles can be defined by the following element:

<!ELEMENT authorization (security-role-name, subject*)>

The security-role-name represents a security role defined for the downstream application. A subject may be a unique user identifier, a group identifier, a special subject, or a security-role-reference name. The security-role-reference name, when specified, means that the security role of a downstream application is assigned to the application.

An authorization table adheres to an application boundary and includes one or more authorization elements, one per security role. Depending on an administrator's preference, the server performs one of two approaches at runtime (step 490, see **Figure 5** and corresponding text for further details regarding role-to-role mapping). The first approach is to keep security-role-reference name assignments and require a user security context to include security role attributes (see **Figures 1**, **3A**, and

corresponding for further details security role attribute inclusion in a security context). The second approach is to expand the security-role-reference name into users, groups, and special subjects based on a downstream server's authorization policy (see **Figures 2, 3B,** and corresponding text for further details regarding subject additions).

In the first approach, a server looks-up security roles that are granted to a specified user by the intended application when authenticating that user. Security role attributes are added into the user's security credentials and are propagated to downstream applications or servers when sending downstream calls to downstream applications. New security roles that are granted to the user by the downstream application are also added to the user's security credentials.

In the second approach, the upstream application's role name that is in a downstream application's authorization table is replaced at runtime by the upstream application's authorization policy (i.e., user, group, special subject assignment). The security-role-binding element in the upstream application indicates that a security runtime should update the downstream application's authorization table whenever the upstream application's user-to-role binding is modified.

To avoid circular dependency, security role equivalency is implemented. If a security role is assigned to another security role and vice versa, then the two security roles are identified as equivalent. Once two security roles are identified as equivalent roles, processing ensures that the two corresponding authorization tables are identical. Role equivalence may also be applied to three or more security roles that are assigned to one another in a circular fashion. For example, role 1 is assigned to role 2, role 2 is assigned to role 3, and role 3 is assigned to role 1. In this example, all three roles are identified as equivalent roles.

Figure 5 is a flowchart showing steps taken in analyzing upstream applications and role-mapping upstream security roles to downstream security roles. Processing commences at **500**, whereupon processing analyzes an upstream application located in upstream application store **515** to identify role-mapping requirements between the upstream application and one or more downstream applications (step **510**). For example, an online banking application allows users to check bank account balances, deposit funds, and withdraw funds. In this example, three security roles are defined in the online banking application which are CheckBalance role, DepositFund role, and WithdrawFund role. In this example, the online banking application (i.e. upstream application) calls a naming space

management application (i.e. downstream application) while executing user requests. In this example, the naming space management application includes four security roles which are CosNamingRead, CosNamingWrite, CosNamingCreate, and CosNamingDelete and in order for a user to perform a balance checking operation that requires the CheckBalance role in the online banking application, the user is also required to have read access to the naming space which requires the CosNamingRole role in the naming space management application. Upstream applications store **515** may be stored on a non-volatile storage area, such as a computer hard drive.

A determination is made as to whether the upstream application includes role-mapping requirements (decision **520**). Processing identifies role-mapping requirements by analyzing extensions included in the upstream application (see **Figure 4** and corresponding text for further details regarding security role extensions).

If the upstream application does not include role-mapping requirements, decision **520** branches to "No" branch **522** whereupon the server loads the upstream application at step **530**, and processing ends at **535**. On the other hand, if the upstream application includes role-mapping requirements, decision **520** branches to "Yes" branch **528** whereupon processing selects a first role-mapping requirement included in the upstream application (step **540**). The role-mapping requirement identifies an upstream security role to be mapped to a downstream security role.

The upstream security role includes upstream subjects (i.e. user identifiers, group identifiers, etc.) which are stored in an upstream authorization table, such as upstream authorization table store **555**. The downstream security role includes downstream subjects which are stored in a downstream authorization table, such as downstream authorization table store **560**. Upstream authorization table store **555** and downstream authorization table store **560** may be stored on a non-volatile storage area, such as a computer hard drive.

Processing role-maps the upstream security role to the downstream security role in one of two ways. The first approach is to keep security-role-reference name assignments and require a user security context (i.e. requests) to include upstream security role identifiers (see **Figures 1, 3A**, and corresponding text for further details regarding security role identifier propagation). The second approach is to expand security-role-reference names into users, groups, and special subjects based on a downstream server's authorization policy (pre-defined process

block **550**, see **Figure 6** and corresponding text for further details regarding role-mapping additions).

A determination is made as to whether the upstream application includes more role-mapping requirements (decision **570**). The upstream application may include multiple role-mapping requirements corresponding to a single downstream application or the upstream application may include multiple role-mapping requirements corresponding to multiple downstream applications. If the upstream application includes more role-mapping requirements, decision **570** branches to "Yes" branch **572** which loops back to select (step **580**) and process the next role-mapping requirement. This looping continues until the upstream application has no more role-mapping requirements to process, at which point decision **570** branches to "No" branch **578** whereupon processing ends at **590**.

Figure 6 is a flowchart showing steps taken in role-mapping upstream security roles to downstream security roles. Processing commences at **600**, whereupon processing selects an upstream authorization table and identifies an upstream security role located in upstream authorization table store **615** (step **610**). The selected upstream security role corresponds to a role-mapping requirement included in an upstream application). Upstream authorization table store **615** may be stored on a non-volatile storage area, such as a computer hard drive. Processing selects a downstream authorization table located in downstream authorization table store **625** which includes a downstream security role corresponding to the role-mapping requirement (step **620**). Downstream authorization table store **625** may be stored on a non-volatile storage area, such as a computer hard drive.

A determination is made as to whether processing is configured to role-map the upstream security role to the downstream security role by either including an upstream security role identifier in the downstream authorization table or by including upstream security role user-to-role mappings in the downstream authorization table (decision **630**) (see **Figures 3A**, **3B**, **4**, and corresponding text for further details regarding role-mapping options). If processing is configured to include an upstream security role identifier in the downstream authorization table, decision **630** branches to "Yes" branch **638** whereupon processing includes an upstream security role identifier in a downstream authorization table entry at step **670**, and processing returns at **680** (see **Figure 3A** and corresponding text for further details regarding upstream security role identifiers).

On the other hand, if processing is configured to include upstream security role assignments (i.e. user identifiers, group identifiers, etc.) in the downstream authorization table, decision **630** branches to "No" branch **632** whereupon processing identifies role assignments corresponding to the upstream security role (step **635**). Processing adds the identified upstream role assignments to the downstream security role at step **640**, and processing returns at **680**.

Figure 7 is a flowchart showing steps taken in a downstream application authorizing a user with an upstream security role assignment. Processing commences at **700**, whereupon the downstream application receives request **715** from upstream application **710** (step **730**). Request **715** includes user identifier **720** and may include roles **725**. User identifier **720** identifies a requesting user and roles **725** includes upstream security role identifiers corresponding to the requesting user's upstream security role assignments.

A determination is made as to whether upstream security roles (e.g. roles **725**) are included in request **715** (decision **740**). If upstream security roles are not included in request **715**, decision **740** branches to "No" branch **742** bypassing upstream role extraction steps. On the other hand, if request **715** includes upstream security roles, decision **740** branches to "Yes" branch **748** whereupon the downstream application extracts the upstream security roles from request **715** (step **750**) and uses the upstream security roles in conjunction with user identifier **720** to determine authorization access (see below for further details).

Processing analyzes request **715** and identifies required downstream security roles for access authorization located in data store **765** (step **760**). Data store **765** may be stored on a non-volatile storage area, such as a computer hard drive. Processing looks-up role assignments corresponding to the required downstream security roles and compares them with user identifier **720** and roles **725** (step **770**). A determination is made as to whether processing matched user identifier **720** or roles **725** (if applicable) with one of the required downstream security role assignments (decision **780**). If processing did not identify a match, decision **780** branches to "No" branch **782** whereupon an error is returned at **785**. On the other hand, if processing identified a match, decision **780** branches to "Yes" branch **788** whereupon processing authorizes the user at step **790**, and processing ends at **795**.

In one embodiment, processing may require information from a second downstream application in order to process the upstream application's

request. In this embodiment, processing includes the matched downstream security role in a second request and send the second request to a second downstream application.

Figure 8 illustrates information handling system 801 which is a simplified example of a computer system capable of performing the invention described herein. Computer system 801 includes processor 800 which is coupled to host bus 805. A level two (L2) cache memory 810 is also coupled to the host bus 805. Host-to-PCI bridge 815 is coupled to main memory 820, includes cache memory and main memory control functions, and provides bus control to handle transfers among PCI bus 825, processor 800, L2 cache 810, main memory 820, and host bus 805. PCI bus 825 provides an interface for a variety of devices including, for example, LAN card 830. PCI-to-ISA bridge 835 provides bus control to handle transfers between PCI bus 825 and ISA bus 840, universal serial bus (USB) functionality 845, IDE device functionality 850, power management functionality 855, and can include other functional elements not shown, such as a real-time clock (RTC), DMA control, interrupt support, and system management bus support. Peripheral devices and input/output (I/O) devices can be attached to various interfaces 860 (e.g., parallel interface 862, serial interface 864, infrared (IR) interface 866, keyboard interface 868, mouse interface 870, and fixed disk (HDD) 872) coupled to ISA bus 840. Alternatively, many I/O devices can be accommodated by a super I/O controller (not shown) attached to ISA bus 840.

BIOS 880 is coupled to ISA bus 840, and incorporates the necessary processor executable code for a variety of low-level system functions and system boot functions. BIOS 880 can be stored in any computer readable medium, including magnetic storage media, optical storage media, flash memory, random access memory, read only memory, and communications media conveying signals encoding the instructions (e.g., signals from a network). In order to attach computer system 801 to another computer system to copy files over a network, LAN card 830 is coupled to PCI bus 825 and to PCI-to-ISA bridge 835. Similarly, to connect computer system 801 to an ISP to connect to the Internet using a telephone line connection, modem 875 is connected to serial port 864 and PCI-to-ISA Bridge 835.

While the computer system described in Figure 8 is capable of executing the invention described herein, this computer system is simply one example of a computer system. Those skilled in the art will

appreciate that many other computer system designs are capable of performing the invention described herein.

One of the preferred implementations of the invention is an application, namely, a set of instructions (program code) in a code module which may, for example, be resident in the random access memory of the computer. Until required by the computer, the set of instructions may be stored in another computer memory, for example, on a hard disk drive, or in removable storage such as an optical disk (for eventual use in a CD ROM) or floppy disk (for eventual use in a floppy disk drive), or downloaded via the Internet or other computer network. Thus, the present invention may be implemented as a computer program product for use in a computer. In addition, although the various methods described are conveniently implemented in a general purpose computer selectively activated or reconfigured by software, one of ordinary skill in the art would also recognize that such methods may be carried out in hardware, in firmware, or in more specialized apparatus constructed to perform the required method steps.

CLAIMS

1. A method of authorizing a client's request at a first downstream application, said method comprising:

receiving, at the first downstream application, a first application request from an upstream application, wherein the first application request is derived from the client's request and includes an upstream security role identifier that was determined by the upstream application;

reading authorization table entries stored in a downstream authorization table accessible from the downstream application;

matching the upstream security role identifier included in the request with at least one of the authorization table entries included in the downstream authorization table; and

authorizing the client's request in response to the matching.

2. The method as described in claim 1 further comprising:

identifying one or more required downstream security roles wherein each authorization table entry corresponds to at least one of the required downstream security roles.

3. The method as described in claim 2 further comprising:

selecting the required downstream security role that corresponds to the matched authorization table entry;

including the selected required downstream security role in a second application request; and

sending the second application request to a second downstream application.

4. The method as described in claim 1 wherein the matched authorization table entry was generated prior to receiving the first application request, the generating further comprising:

receiving a role-mapping requirement wherein the role-mapping requirement includes an upstream security role and a downstream security

role, the upstream security role corresponding to the upstream security role identifier;

including the upstream security role identifier in the matched authorization table entry, the matched authorization table entry corresponding to the downstream security role.

5. The method as described in claim 4 further comprising:

determining whether the upstream security role and the downstream security role are equivalent; and

configuring the upstream security role and the downstream security role such that they are equivalent.

6. A method of expanding an upstream security role to include a downstream application, said method comprising:

receiving a security role-mapping request that corresponds to an upstream application and the upstream security role;

selecting a downstream security role corresponding to the security role-mapping request and the downstream application; and

adding one or more downstream authorization table entries to the downstream security role, wherein the authorization table entries correspond to the upstream security role.

7. The method as described in claim 6 wherein at least one of the authorization table entries is selected from the group consisting of a user identifier, a group identifier, and an upstream security role identifier.

8. The method as described in claim 6 further comprising:

determining whether the upstream security role and the downstream security role are equivalent; and

configuring the downstream authorization table entries and one or more upstream authorization table entries such that they are equivalent, the upstream authorization table entries corresponding to the upstream security role.

9. The method as described in claim 6 further comprising:

receiving an application request from the upstream application, wherein the request includes a user identifier;

matching the user identifier with one or more of the downstream authorization table entries; and

authorizing the application request in response to the matching.

10. An information handling system comprising:

one or more processors;

a memory accessible by the processors;

one or more nonvolatile storage devices accessible by the processors; and

a client authorization tool to authorize a client's request at a first downstream application, the client authorization tool including: means for receiving, at the first downstream application, a first application request from an upstream application, wherein the first application request is derived from the client's request and includes an upstream security role identifier that was determined by the upstream application;

means for reading authorization table entries stored in a downstream authorization table accessible from the downstream application, the downstream authorization table located in one of the nonvolatile storage devices;

means for matching the upstream security role identifier included in the request with at least one of the authorization table entries included in the downstream authorization table; and

means for authorizing the client's request in response to the matching.

11. The information handling system as described in claim 10 further comprising:

means for identifying one or more required downstream security roles located in one of the nonvolatile storage devices wherein each authorization table entry corresponds to at least one of the required downstream security roles.

12. The information handling system as described in claim 11 further comprising:

means for selecting the required downstream security role that corresponds to the matched authorization table entry;

means for including the selected required downstream security role in a second application request; and

means for sending the second application request to a second downstream application.

13. The information handling system as described in claim 10 wherein the matched authorization table entry was generated prior to receiving the first application request, the generating further comprising:

means for receiving a role-mapping requirement wherein the role-mapping requirement includes an upstream security role and a downstream security role, the upstream security role corresponding to the upstream security role identifier;

means for including the upstream security role identifier in the matched authorization table entry, the matched authorization table entry corresponding to the downstream security role.

14. The information handling system as described in claim 13 further comprising:

means for determining whether the upstream security role and the downstream security role are equivalent; and

means for configuring the upstream security role and the downstream security role such that they are equivalent.

15. A computer program product stored on a computer operable media for authorizing a client's request at a first downstream application, said

computer program product comprising means for carrying out a method according to any one of claims 1 to 6.

5