



- (51) International Patent Classification:
H04L 29/06 (2006.01) H04L 29/08 (2006.01)
G06F 21/62 (2013.01)
- (21) International Application Number:
PCT/EP2014/071228
- (22) International Filing Date:
3 October 2014 (03.10.2014)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
2979/DEL/2013 7 October 2013 (07.10.2013) IN
- (71) Applicant: ALCATEL LUCENT [FR/FR]; 148/152 route de la Reine, F-92100 Boulogne-Billancourt (FR).
- (72) Inventor: DWIVEDI, Abhishek; #727, 11th B Cross, 25th Main, HSR Layout, Karanataka, Bangalore 560102 (IN).
- (74) Agent: SARUP, David Alexander; ALCATEL LUCENT TELECOM LTD, Intellectual Property Business Group, Christchurch Way, Greenwich, London Greater London SE10 0AG (GB).

- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

(54) Title: SYSTEMS AND METHODS FOR COMMAND EXECUTION AUTHORIZATION

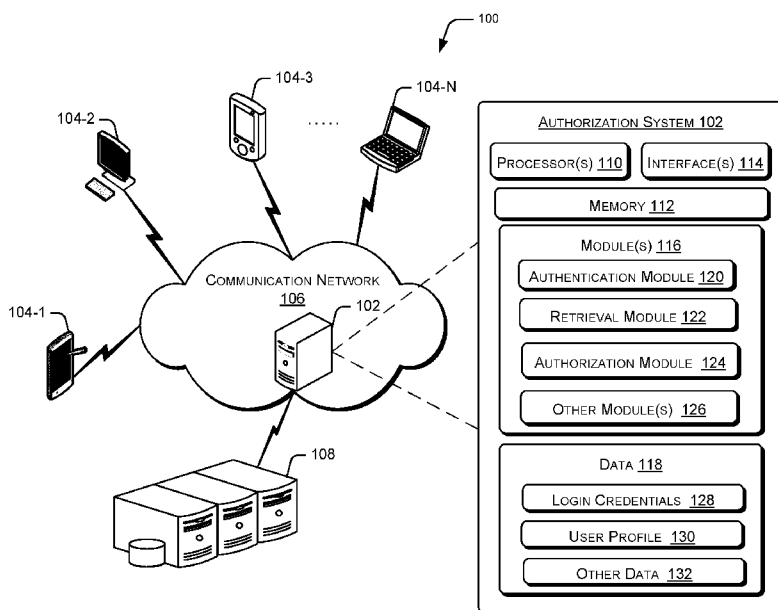


Figure 1

(57) Abstract: Method(s) and system(s) for authorization of execution of a command on an authorization system (102) are described. The method includes transmitting an access request received from a user to a LDAP server (108). The access request includes login credentials of the user. Based on the login credentials, a user profile is retrieved from the LDAP server (108), once the user is authenticated. The user profile includes a set of commands associated with the user. The method includes storing the user profile in a cache memory of the authorization system (102). The method includes receiving the command from the user and comparing the command with the set of commands stored in the cache memory. Based on the comparison, it is determined whether the command is authorized for execution on the authorization system (102). Further, based on the determination, the user is authorized to execute the command on the authorization system (102).

SYSTEMS AND METHODS FOR COMMAND EXECUTION AUTHORIZATION

FIELD OF INVENTION

[0001] The present subject matter relates to authorization of execution commands and, particularly, but not exclusively, to authorization of execution of a command on a remote server.

BACKGROUND

[0002] In today's digital age, enterprises are increasingly leveraging communication networks to enable employees to work remotely from anywhere at anytime. While providing access to the various enterprise resources, the enterprises typically implement network security to prohibit any unauthorized user from accessing the network. For this, generally a security architecture referred as authentication, authorization and accounting (AAA) is used. AAA enables control over access by users, such as which users have access to which services, and how much of the resources they may use.

[0003] As is well known, the AAA architecture authenticates a user to verify the user's identity. In case the authentication fails, the AAA architecture does not allow the user to access the network resources. On the other hand, if authentication is successful, the user is authorized to access network resources. Generally, Terminal Access Controller Access-Control System Plus (TACACS+) and Remote Access Dial In User Service (RADIUS) are widely used network protocols that provide AAA functionality.

SUMMARY

[0004] This summary is provided to introduce concepts related to authorization of execution of a command on a remote server. This summary is not intended to identify essential features of the claimed subject matter nor is it directed to use in determining or limiting the scope of the claimed subject matter.

[0005] In an implementation, a method for authorization of execution of a command on an authorization system is described. The method includes transmitting an access request received from a user to a lightweight directory access protocol (LDAP) server. The access request may

include login credentials of user for authenticating the user to access the authorization system. Further, the method includes retrieving, based on the login credentials, a user profile from the LDAP server, once the user is authenticated. The user profile includes a set of commands associated with the user. The set of commands is one of a set of allowed commands and a set of denied commands. The method may also include storing the user profile in a cache memory of the authorization system. Furthermore, the method includes receiving the command to be executed on the authorization system. The method includes comparing the command received from the user with the set of commands stored in the cache memory of the authorization system. The method further comprises determining, based on the comparison, whether the command is authorized for execution on the authorization system. Further, the method may include authorizing the user, based on the determination, to execute the command on the authorization system.

[0006] In another implementation, an authorization system for authorization of execution of a command is described. The authorization system includes a processor, an authentication module coupled to the processor, a retrieval module coupled to the processor, and an authorization module coupled to the processor. The authentication module receives an access request from the user. The access request includes login credentials of the user for authenticating the user to access the authorization system. The authentication module transmits the access request received from the user to a lightweight directory access protocol (LDAP) server for authentication. Further, upon authentication, the retrieval module retrieves a user profile from the LDAP server. The user profile includes a set of commands associated with the user. The set of commands is one of a set of allowed commands and a set of denied commands. The retrieval module may store the user profile in a cache memory of the authorization system. Thereafter, the authorization module receives, from the user, the command to be executed on the authorization system. The authorization module compares the command received from the user with the set of commands stored in the cache memory of the authorization system. Based on the authorization, the authorization module determines whether the command is authorized for execution on the authorization system. Further, based on the determination, the authorization module authorizes the user to execute the command on the authorization system.

[0007] In yet another implementation, a non-transitory computer-readable medium having embodied thereon a computer program for executing a method for authorization of execution of a

command on an authorization system is described. The method includes transmitting an access request received from a user to a lightweight directory access protocol (LDAP) server. The access request may include login credentials of user for authenticating the user to access the authorization system. Further, the method includes retrieving, based on the login credentials, a user profile from the LDAP server, once the user is authenticated. The user profile includes a set of commands associated with the user. The set of commands is one of a set of allowed commands and a set of denied commands. The method may also include storing the user profile in a cache memory of the authorization server. Furthermore, the method includes receiving the command to be executed on the authorization server. The method includes comparing the command received from the user with the set of commands stored in the cache memory of the authorization system. The method further comprises determining, based on the comparison, whether the command is authorized for execution on the authorization server. Further, the method may include authorizing the user, based on the determination, to execute the command on the authorization system.

BRIEF DESCRIPTION OF THE FIGURES

[0008] The detailed description is described with reference to the accompanying figures. In the figures, the left-most digit(s) of a reference number identifies the figure in which the reference number first appears. The same numbers are used throughout the figures to reference like features and components. Some embodiments of system and/or methods in accordance with embodiments of the present subject matter are now described, by way of example only, and with reference to the accompanying figures, in which:

[0009] Fig. 1 illustrates a network implementation of an authorization system, in accordance with an embodiment of the present subject matter;

[0010] Fig. 2 shows a flowchart illustrating an exemplary method for authorization of execution of a command on a remote server, in accordance with an embodiment of the present subject matter.

[0011] It should be appreciated by those skilled in the art that any block diagrams herein represent conceptual views of illustrative systems embodying the principles of the present subject matter. Similarly, it will be appreciated that any flow charts, flow diagrams, state transition diagrams, pseudo code, and the like represent various processes which may be

substantially represented in computer readable medium and so executed by a computer or processor, whether or not such computer or processor is explicitly shown.

DESCRIPTION OF EMBODIMENTS

[0012] Systems and methods for authorization of execution of a command on a remote server are described. The systems and methods can be implemented in a variety of communication devices. The communication devices that can implement the described method(s) include, but are not limited to, mobile phones, hand-held devices, laptops or other portable computers, personal digital assistants (PDAs), notebooks, tablets, and the like.

[0013] With the growing popularity of wireless networks, both for personal use and business use, one of the most difficult jobs a network administrator faces in today's network security environment is limiting access of network services to authorized users. Equally challenging for the network administrator is closely monitoring what services are used and when, and the frequency of use by users. In this respect, AAA architecture has been developed to facilitate control over the access rights of users, such as which users have access to which services, and how much of the resources they may use. Typically, TACACS+ protocol, implementing the AAA architecture, is deployed in enterprises to restrict access by unauthorized users. To do so, information pertaining to different users is stored in a TACACS+ server. The information pertaining to the users typically includes user credentials as well as commands that the users are authorized to execute on a remote server, such as a network access server (NAS).

[0014] In this respect, when a user sends a command for execution at the NAS, the NAS communicates with the TACACS+ server for authenticating the user. The TACACS+ server authenticates the user by comparing the login details of the user with the user credentials stored therein. Once authenticated, for every command, the NAS sends an authorization request to the TACACS+ server. The TACACS+ server accesses the information pertaining to the user to determine whether or not the user has authorization to perform the command. Thereafter, the TACACS+ server may send a transmission back to the remote server via TACACS+ protocol indicating the results of the determination regarding authorization. However, there may be instances where, while sending and receiving authorization request and/or determination, connectivity between the TACACS+ server and the NAS is lost. This may result in the user not

being able to execute the commands after the user has been authenticated by the NAS and even when the commands are authorized to the user.

[0015] To overcome the above mentioned drawback, RADIUS protocol is used for authorization of the commands. In this scenario, for executing any command, the NAS may send an authorization request to a RADIUS server. Based on the authorization request, a user profile may be retrieved from the RADIUS server and is transferred to the NAS. The user profile includes information regarding the commands that the user is authorized to execute. Each time the user attempts to execute a command at the remote server, the user profile may be referred to. Accordingly, the command may then be authorized or denied based on the results of this reference. However, the RADIUS protocol does not allow performing authentication and command authorization operations independently of each other. This may become inconvenient for the network administrators, as the network administrators may be restricted to use only the RADIUS protocol for authentication and authorization purposes.

[0016] In addition, even when the number of executable commands is more, the above described approach only provides a method of identifying a particular command or set of commands that are executable by a particular user. For example, a user may not be authorized to execute a few commands from a set of commands. As per the RADIUS protocol, the user profile will include only those commands that the user is authorized to execute. This may be cumbersome in cases where the user is denied a couple of commands from a huge set of commands, as the NAS will parse through the list of allowed commands to determine if the command is denied to the user or not. Considering a scenario where out of 1000 commands, the user is denied 50 commands. According to the above approach, the user profile in the RADIUS server includes a list of 950 commands that are authorized to the user. If the user attempts to execute a command, which is denied to the user, the NAS needs to parse through all 950 commands, to come to a decision of denying the execution of that command. This may become an overhead for the NAS and slow down the command executions

[0017] The present subject matter describes systems and methods for authorization of execution of a command on a remote server. In an implementation, the systems and methods as described herein implements a lightweight directory access protocol (LDAP) for authorizing commands. The LDAP may be used as an authentication, authorization, and accounting (AAA) protocol. Further, the LDAP may perform command authorization independent of authentication.

For example, the network administrator may use TACACS+ protocol for authenticating the user and may further implement LDAP for authorizing the user. Accordingly, the present subject matter provide flexibility to the network administrators to use LDAP for either authentication or authorization or for both. Further, the systems and methods as described herein implements an LDAP server configured to store a user profile therein. The user profile may include information about the user, such as login credentials of the user in addition to a set of commands associated with a user. In an implementation, the set of commands may either be a set of allowed commands or a set of denied commands. In an implementation, the user profile may be stored in schema of the LDAP server in the form of attributes. An attribute may provide a description of a component of the LDAP server. Entries in the LDAP server may be made up of attributes which consist of an attribute type and one or more values. In an implementation, the value for each of the attributes may be defined by using extended regular expression (ERE).

[0018] In addition to associating a particular set of commands with the user, a command deny flag may be defined in the schema of the LDAP server. The command deny flag may have a single value, such as ON or OFF. The command deny flag may facilitate in indicating whether a particular set of commands is authorized or denied to the user. For example, when the value of the command deny flag is OFF for a particular set of commands, it may be understood that the user is authorized to execute the commands as listed in that particular set. Referring to the above-mentioned example in which from a total of 1000 commands, the user is denied 50 commands. The present subject matter may facilitate the network administrator to explicitly associate the 50 commands with the user profile. Accordingly, the network administrator may list the 50 commands in the user profile and may associate the command deny flag, having the value as ON, for the set of commands. This may indicate that the apart from those 50 commands the user is authorized to execute any other command. Accordingly, the present subject matter facilitates the network administrator to explicitly associate a set of any one of authorized or denied commands with the user. This may facilitate in reducing processing time while determining authorization of a particular command.

[0019] To execute a particular command, the user may send an access request to a remote server, such as the NAS. In an implementation, an LDAP session may be established between the user and the NAS when the user sends the access request to the NAS. Further, the access request may include login credentials, such as user identifier and password for the accessing the NAS.

The NAS may share the login credentials with the LDAP server to determine whether or not the user is authenticated to access the NAS. Accordingly, the LDAP server may compare the login credentials as provided by the user in the access request with the login credentials stored in the LDAP server. Once authenticated, the user may attempt to execute a command on the NAS. In this respect, the NAS may send an authorization request to the LDAP. Based on the authorization request, the user profile may be retrieved from the LDAP server. In an implementation, the user profile may be retrieved from the LDAP server as soon as the user is authenticated by the NAS. Further, the user profile may be stored in a cache memory of the NAS. This may eliminate the need the need to send authorization requests to the LDAP server every time the user wishes to execute a command on the NAS.

[0020] As described above, the user profile may include the set of commands, either allowed commands or denied commands, associated with the user. Accordingly, when the user may attempt to execute the command on the NAS, the NAS may identify availability of the command in the set of commands associated with the user. In addition, the NAS may also determine the value of the command deny flag associated with the set of commands. Based on the identification, the NAS may determine whether the user is authorized to execute the command on the NAS. In an example, if the command is listed in the set of commands defined for the user and the value of the command deny flag is OFF, the NAS may authorize the execution of the command. In an alternative example, if the command is not listed in the set of commands defined for the user and the value of the command deny flag is OFF, the NAS may deny the user from executing the command. As may be seen, the present subject matter facilitates the network administrator to associate either the set of allowed commands or the set of denied commands, with the user.

[0021] Accordingly, the present subject matter utilizes LDAP that facilitates separation of the authentication phase from the authorization phase. This may provides the network administrator with an option of using different authentication protocols, such as LDAP, TACACS+, and RADIUS, to authenticate the user and to determine that user's level of access. Further, the present subject matter facilitates network administrators to explicitly allow or deny some commands from a set of commands thereby enabling faster processing of the authorization requests. In addition, as the user details are cached in the remote server, the user may execute

commands that are authorized to the user, even when the LDAP server is not connected to the remote server.

[0022] The above methods and system are further described in conjunction with the following figures. It should be noted that the description and figures merely illustrate the principles of the present subject matter. It will thus be appreciated that those skilled in the art will be able to devise various arrangements that, although not explicitly described or shown herein, embody the principles of the present subject matter and are included within its spirit and scope. Furthermore, all examples recited herein are principally intended expressly to be only for pedagogical purposes to aid the reader in understanding the principles of the present subject matter and the concepts contributed by the inventor(s) to furthering the art, and are to be construed as being without limitation to such specifically recited examples and conditions. Moreover, all statements herein reciting principles, aspects, and embodiments of the present subject matter, as well as specific examples thereof, are intended to encompass equivalents thereof.

[0023] It will also be appreciated by those skilled in the art that the words during, while, and when as used herein are not exact terms that mean an action takes place instantly upon an initiating action but that there may be some small but reasonable delay, such as a propagation delay, between the initial action, and the reaction that is initiated by the initial action. Additionally, the words “connected” and “coupled” are used throughout, for clarity of the description and can include either a direct connection or an indirect connection.

[0024] The manner in which the systems and the methods for providing access control based on command authorization in a communication network has been explained in detail with respect to the Figures 1-2. While aspects of described systems and methods for providing access control based on command authorization in a communication network can be implemented in any number of different computing systems, transmission environments, and/or configurations, the embodiments are described in the context of the following exemplary system(s).

[0025] Fig. 1 illustrates a network environment 100 implementing an authorization system 102, in accordance with an embodiment of the present subject matter. In said embodiment, the network environment 100 includes the authorization system 102 configured to authorize execution of a command on a remote server.

[0026] In one implementation, the network environment 100 may be a company network, including thousands of office personal computers, laptops, various servers, such as blade servers, and other computing devices. In another implementation, the network environment 100 may be a smaller private network. In yet another implementation, the network environment 100 may be a public network, such a public cloud.

[0027] The authorization system 102 of the present subject matter may be implemented in a network server, such as a network access server (NAS). The NAS may be understood as a computer server that enables an independent service provider (ISP) to provide users with Internet access. In one implementation, the authorization system 102 may be included within an existing information technology infrastructure or a database management structure. Further, it will be understood that the authorization system 102 may be connected to a plurality of user devices 104-1, 104-2, 104-3,...,104-N, collectively referred to as the user devices 104 or individually as a user device 104. The user device 104 may include, but is not limited to, a desktop computer, a portable computer, a mobile phone, a handheld device and a workstation. The user devices 104 may be used by users, such as database analysts, programmers, developers, data architects, software architects, module leaders, projects leaders, database administrator (DBA), stakeholders, and the like.

[0028] As shown in the figure, the user devices 104 are communicatively coupled to the authorization system 102 over a communication network 106 through one or more communication links for facilitating one or more end users to access and operate the authorization system 102. In an implementation, the user devices 104 may establish a communication session with the authorization system 102 through a protocol, such as a Lightweight Directory Access Protocol (LDAP). The communication network 106 may be a wireless network, a wired network, or a combination of wired and wireless network.

[0029] The communication network 106 can be a collection of individual networks, interconnected with each other and functioning as a single large network (e.g., the internet or an intranet). Examples of such individual networks include, but are not limited to, Global System for Mobile Communication (GSM) network, Universal Mobile Telecommunications System (UMTS) network, Personal Communications Service (PCS) network, Time Division Multiple Access (TDMA) network, Code Division Multiple Access (CDMA) network, Next Generation Network (NGN), IP-based network, Public Switched Telephone Network (PSTN), Integrated

Services Digital Network (ISDN), Long Term Evolution (LTE), Passive Optical Network, and the like. The communication network 106 may either be a dedicated network or a shared network, which represents an association of the different types of networks that use a variety of protocols, for example, Hypertext Transfer Protocol (HTTP), Transmission Control Protocol/Internet Protocol (TCP/IP), etc., to communicate with each other. Further, depending on the technology, the communication network 106 may include various network entities, such as gateways, routers; however, such details have been omitted for ease of understanding.

[0030] Further, the network environment 100 may include an authorization server, such as an Lightweight Directory Access Protocol (LDAP) server 108. The LDAP server 108 may be communicatively coupled to the authorization system 102. The LDAP server 108 may include a schema, which may be customized by an administrator, by adding new elements or by updating existing elements thereto. Further, the schema may include entries associated with different objects. In an implementation, the schema of the LDAP server 108 may store user profiles, such that information (entries) about the users (objects) is stored in the form of attributes and object classes. Each object class may be associated with a plurality of attributes. An attribute may provide a description of a component of the LDAP server. In an example, the LDAP server 108 may be configured to store each user as an object class. Further, the plurality of attributes associated with a user may hold specific data about the user, such as a name, an address, or a telephone number. The user profile may include a set of commands that may be associated with the user. The set of commands may be stored in the schema of the LDAP server 108 as attributes.

[0031] In an implementation, each of the plurality of attributes may be associated with one or more values, in the LDAP server 108. In the present implementation, the value for each of the attributes may be defined by using extended regular expression (ERE). Further, each of the plurality of attributes may be associated with one or more values. Accordingly, multiple ERE's may be associated with each attribute. In addition, a command deny flag may be defined in the schema of the LDAP server 108. The command deny flag may have a single value, such as ON and OFF. The command deny flag may facilitate in explicitly indicating whether a particular set of commands is authorized or denied to the user.

[0032] Example 1 provides an exemplary schema that defines an attribute 'commandAttribute'. This attribute defines those commands that may be allowed or denied based on values of another attribute, i.e., 'commandDenyFlag'. The commands that may not be defined

in the 'commandAttribute' may be understood to be either allowed or denied based on the value of the 'commandDenyFlag'.

Example 1: Command authorization allow attribute

```

attributetype ( CommandAuthorLdapAtt:1
    NAME 'commandAttribute'
    DESC 'Commands which user is authorized/unauthorized to execute'
    EQUALITY caseIgnoreMatch
    SUBSTR caseIgnoreSubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.115
)

```

[0033] Further, Example 2 provides an exemplary schema that defines an attribute 'commandDenyFlag'. As discussed earlier, this attribute is single valued and based on the value assigned to 'commandDenyFlag', commands may be either allowed or denied to the user.

Example 2: Command authorization deny attribute

```

attributetype ( CommandAuthorLdapAtt:2
    NAME 'commandDenyFlag'
    DESC 'This attribute decides whether the commands present in
'commandAttribute' are allowed or denied'
    EQUALITY caseIgnoreMatch
    SUBSTR caseIgnoreSubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.115
    SINGLE VALUE
)

```

[0034] In addition, Example 3 provides an exemplary schema definition of an object class. The below example indicates the object class as 'CommandAuthorUser'

Example 3: Object class attribute

```

objectclass ( CommandAuthorLdap:1
    NAME 'CommandAuthorUser'
    DESC 'User with command authorization'
)

```

```

SUP person
STRUCTURAL
MUST (
    uid $ userPassword )
MAY (
    commandAttribute $ commandDenyFlag)
)

```

[0035] Again referring to Fig. 1, the authorization system 102 includes one or more processor(s) 110, and a memory 112 connected to the processor 110. The processor 110 may include microprocessors, microcomputers, microcontrollers, digital signal processors, central processing units, state machines, logic circuitries and/or any other devices that manipulate signals and data based on operational instructions. The processor 110 can be a single processing unit or a number of units, all of which could also include multiple computing units. Among other capabilities, the processor 110 is configured to fetch and execute computer-readable instructions stored in the memory 204.

[0036] Functions of the various elements shown in the figures, including any functional blocks labeled as “processor(s)”, may be provided through the use of dedicated hardware as well as hardware capable of executing software in association with appropriate software. When provided by a processor, the functions may be provided by a single dedicated processor, by a single shared processor, or by a plurality of individual processors, some of which may be shared. Moreover, explicit use of the term “processor” should not be construed to refer exclusively to hardware capable of executing software, and may implicitly include, without limitation, digital signal processor (DSP) hardware, network processor, application specific integrated circuit (ASIC), field programmable gate array (FPGA), read only memory (ROM) for storing software, random access memory (RAM), and non volatile storage. Other hardware, conventional and/or custom, may also be included.

[0037] The memory 112 can include any computer-readable medium known in the art including, for example, volatile memory, such as RAM and/or non-volatile memory, such as flash. Further, the authorization system 102 includes one or more interface(s) 114. The interface 114 may include a variety of software and hardware interfaces, for example, interfaces for peripheral device(s), such as data input output devices, referred to as I/O devices, storage

devices, network devices, etc. The I/O device(s) may include Universal Serial Bus (USB) ports, Ethernet ports, host bus adaptors, etc., and their corresponding device drivers.

[0038] The interface 114 may facilitate the communication between the authorization system 102 and various communication and computing devices and various networks, such as Global System for Mobile Communication (GSM) network, Universal Mobile Telecommunications System (UMTS) network, Personal Communications Service (PCS) network, Time Division Multiple Access (TDMA) network, Code Division Multiple Access (CDMA) network, Next Generation Network (NGN), IP-based network, Public Switched Telephone Network (PSTN), Integrated Services Digital Network (ISDN), networks that use a variety of protocols, for example, Hypertext Transfer Protocol (HTTP), Transmission Control Protocol/Internet Protocol (TCP/IP), Wireless Application Protocol (WAP).

[0039] The authorization system 102 may include module(s) 116 and data 118. The module(s) 116 include routines, programs, objects, components, data structures, etc., which perform particular tasks or implement particular abstract data types. The modules 116 may also be implemented as, signal processor(s), state machine(s), logic circuitries, and/or any other device or component that manipulate signals based on operational instructions.

[0040] Further, the modules 116 can be implemented in hardware, instructions executed by a processing unit, or by a combination thereof. The processing unit can comprise a computer, a processor, such as the processor 110, a state machine, a logic array or any other suitable devices capable of processing instructions. The processing unit can be a general-purpose processor which executes instructions to cause the general-purpose processor to perform the required tasks or, the processing unit can be dedicated to perform the required functions.

[0041] In another aspect of the present subject matter, the modules 116 may be machine-readable instructions (software) which, when executed by a processor/processing unit, perform any of the described functionalities. The machine-readable instructions may be stored on an electronic memory device, hard disk, optical disk or other machine-readable storage medium or non-transitory medium. In one implementation, the machine-readable instructions can be also be downloaded to the storage medium via a network connection.

[0042] In one implementation, the module(s) 116 of the authorization system 102 may include an authentication processing module 120, a retrieval module 122, an authorization module 124, and other module(s) 126. The other module(s) 126 may include programs or coded instructions that supplement applications and functions of the authorization system 102. It will be evident that the modules 116 of the authorization system 102 are implemented in a network access server. It will be evident that the module(s) 116 and data 118 may be a part of the memory 112 of the authorization system 102. On the other hand, the data 118, amongst other things, serves as a repository for storing data processed, received, associated, and generated by one or more of the module(s) 116.

[0043] The data 118 includes, for example, login credentials 128, user profile 130, and other data 132. The other data 132 includes data generated as a result of the execution of one or more modules in the other module(s) 126.

[0044] As described above, the authorization system 102 facilitates in providing command authorization to a user. The authorization system 102 may authenticate the user based on the login credentials received from the user. Once authenticated, the user may attempt to execute a command on the authorization system 102. The authorization system 102, upon receiving a command from the user, retrieves the user profile from the LDAP server 108. This user profile is stored in a cache memory of the authorization system 102. The authorization system 102 may compare the command with the list of commands in the user profile and based on the comparison, authorize or deny the execution of the command by the user.

[0045] In an implementation, the authentication module 120 may establish a session with the user. The authentication module 120 may facilitate the user to connect with the authorization system 102 by establishing the session through any one of existing authentication protocols, such as TACACS+, RADIUS, Local, and Kerberos. Accordingly, the user may connect with the authorization system 102 through any authentication protocol. Once the session is established, the authentication module 120 may receive an access request from the user to gain access of the authorization system 102. The access request may include details, such as login credentials 128, of the user. Upon receiving the access request, the authentication module 120 may share the login credentials 128 of the user with the LDAP server 108 for confirming the validity of the user. The LDAP server 108 may compare the login credentials 128 received from the user with

the login credentials stored therein. In an implementation, the authentication module 120 may authenticate the user through any one of the existing authentication protocols, such as TACACS+, RADIUS, Local, and Kerberos. Further, when the LDAP server 108 confirms the validity of the user, as an authenticated user, the user may get access of the authorization system 102, else the authentication module 120 may deny the access to the user.

[0046] In an implementation, the authorization system 102 may facilitate the network administrator to enable or disable command authorization. Accordingly, as soon as the user is authenticated by the authentication module 120, the retrieval module 122 may determine whether command authorization, through the LDAP server 108 is enabled or not. If the command authorization is enabled through the LDAP server 108, the retrieval module 122 may retrieve the user profile 130 from the LDAP server 108 as soon as the user is authenticated by the authentication module 120. In another implementation, the retrieval module 122 may retrieve the user profile 130 from the LDAP server 108 on receiving a command for execution from the user. As described earlier, the user profile may include a set of commands associated with the user. The set of commands may be a set of allowed commands or a set of denied commands. The user profile may also include a command deny flag associated with the set of commands associated with the user. The command deny flag is associated with a value, based on which the authorization of command execution is decided for the user. It will be understood by a person skilled in the art that the user profiles may be defined by administrators. The LDAP server 108 may provide flexibility to the administrator to associate either the set of commands that is allowed to the user or the set of commands that is denied to the user. This may facilitate in reducing the time taken in parsing through the set of allowed commands, when the denied commands are less, before authorizing or denying a user to execute a command. For example, the administrator may associate the set of denied commands with the user, in the LDAP server 108, thereby indicating that the commands falling outside the set of denied commands are allowed to the user.

[0047] In addition, the authorization system 102 may facilitate the user to enable or disable the command authorization through the LDAP server 108. In a scenario where the command authorization is disabled by the network administrator, as soon as the user is authenticated, the user may execute a command on the authorization system 102 as implemented on the remote server. In another scenario where the command authorization is disabled by the

network administrator, any command sent by the user is authorized by the authorization system 102 before the user may execute the command. In this respect, the retrieval module 122 may store the user profile 130 in the cache memory of the authorization system 102. The user profiles 130 are stored in the LDAP server 108; however, it will be evident to a person skilled in the art that the user profiles 130 may be maintained in a database that may be external to the LDAP server 108.

[0048] Once authenticated, the user may attempt to execute a command on the authorization system 102. In an implementation, the command may be received by the authorization module 124. In an implementation, the command sent by the user may be defined as:

```
dn: cn=abhi,dc=my-domain,dc=com
objectClass: commandAuthorUser
cn: abhi
uid: abhi
userPassword: welcome
commandAttribute: telnet .*
commandAttribute: show aaa .*
```

[0049] Accordingly, the authorization module 124 may ascertain availability of the command in the set of commands associated with the user. Further, the authorization module 124 may check the value of the command deny flag that may be associated with the set of commands. In an implementation, the command deny flag associated with the above command may be defined as:

```
commandDenyFlag: OFF
description: User can telnet and execute only 'show aaa' commands
```

[0050] Based on the description provided for the command deny flag associated with the user profile, the authorization module 124 may allow user “abhi” to telnet to anything and only execute the “show aaa.*” commands. As per the user profile defined by an administrator, remaining all commands are denied to the user “abhi”.

[0051] In another implementation, the command sent by a user may be defined as:

```
dn: cn=ravi,dc=my-domain,dc=com
objectClass: commandAuthorUser
cn: ravi
    uid: ravi
userPassword: notwelcome
commandAttribute: ping .*
commandAttribute: show configuration snapshot
commandDenyFlag: ON
description: User cannot ping and execute only 'show configuration
snapshot' command, rest of the commands it will be allowed
```

[0052] As mentioned above, the authorization module 124 may deny user “ravi” to ping and execute 'show configuration snapshot' command, rest of the commands will be allowed to the user. Further, storing the user profiles 130 in the cache memory of the authorization system 102 may eliminate the need to connect to the LDAP server 108 every time the user attempts to execute a command.

[0053] The authorization system 102 of the present subject matter utilizes LDAP that facilitates separation of the authentication phase from the authorization phase. This may provide the network administrator with an option of using different authentication protocols to authenticate the user and to determine that user's level of access. Further, the present subject matter facilitates network administrators to associate those either allowed or denied commands from a set of commands, with the user profile, thereby enabling faster processing of the authorization requests. In addition, as the user details are cached in the remote server, the user may execute commands that are authorized to the user, even when the LDAP server 108 is not connected to the remote server.

[0054] Fig. 2 illustrates a method 200 for authorization of execution of a command on a remote server, according to an embodiment of the present subject matter. The order in which the method is described is not intended to be construed as a limitation, and any number of the described method blocks can be combined in any order to implement the method 200 or any alternative method. Additionally, individual blocks may be deleted from the method without departing from the spirit and scope of the subject matter described herein. Furthermore, the

method can be implemented in any suitable hardware, software, firmware, or combination thereof.

[0055] The method(s) may be described in the general context of computer executable instructions. Generally, computer executable instructions can include routines, programs, objects, components, data structures, procedures, modules, functions, etc., that perform particular functions or implement particular abstract data types. The methods may also be practiced in a distributed computing environment where functions are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, computer executable instructions may be located in both local and remote computer storage media, including memory storage devices.

[0056] A person skilled in the art will readily recognize that steps of the method(s) 200 can be performed by programmed computers. Herein, some embodiments are also intended to cover program storage devices or computer readable medium, for example, digital data storage media, which are machine or computer readable and encode machine-executable or computer-executable programs of instructions, where said instructions perform some or all of the steps of the described method. The program storage devices may be, for example, digital memories, magnetic storage media, such as a magnetic disks and magnetic tapes, hard drives, or optically readable digital data storage media. The embodiments are also intended to cover both communication network and communication devices to perform said steps of the method(s).

[0057] At block 202, the method 200 may include establishing a session with the user. In an implementation, the authentication module 120 may establish the session with the user. The session may facilitate the user to connect to the authorization system 102 implemented in the remote server, such as NAS. In the present implementation, the session may be an LDAP session, however, the session may be established through any authentication protocol, such as RADIUS, TACACS+, and Local. Once the session is established, the user may send an access request to the authorization system 102. The access request may be intended to gain access of the authorization system 102..

[0058] At block 204, the method 200 may include authenticating the user based on the access request. In an implementation, the authentication module 120 may authenticate the user through any one of the existing authentication protocols, such as TACACS+, RADIUS, Local,

and Kerberos. The access request may include login credentials 128 of the user. The authentication module 120 may share the login credentials 128 of the user with the LDAP server 108 to confirm validity of the user. Once a confirmation is received from the LDAP server 108, the authentication module 120 may authenticate the user and allow access to the authorization system 102.

[0059] At block 206, the method 200 may include determining whether command authorization, through the LDAP server 108 is enabled or not. In an implementation, the retrieval module 122 may determine enablement of the command authorization. The authorization system 102 may facilitate the network administrator to enable or disable command authorization. If the command authorization is not enabled through the LDAP server 108, the method 200 moves to step 208.

[0060] At block 208, the method 200 may include receiving a command from the user. In an implementation, the authorization module 124 may receive the command from the user.

[0061] At block 210, the method 200 may include executing the command on the authorization system 102. In an implementation, the authorization module 124 may execute the command on the authorization system 102. Further, when the command authorization is enabled through the LDAP server 108, the method 200 moves to step 212.

[0062] At block 212, the method 200 may include retrieving a user profile from the LDAP server 108. In an implementation, the retrieval module 122 may retrieve the user profile from the LDAP server 108. The user profile may include a set of commands associated with the user. The set of commands may be a set of allowed commands or a set of denied commands. In addition, the user profile may include a command deny flag associated with the set of commands. The command deny flag may be indicative of whether or not the user is authorized to execute the set of commands associated therewith.

[0063] At block 214, the method 200 may include storing the user profile in a cache memory of the authorization system 102. In an implementation, the retrieval module 122 may store the user profile in the cache memory of the authorization system 102.

[0064] At block 216, the method 200 may include receiving the command from the user for execution on the authorization system 102. In an implementation, the authorization module 124 may receive the command from the user.

[0065] At block 218, the method 200 may include comparing the command received from the user with the set of commands stored in the cache memory. In an implementation, the authorization module 124 may compare the command with the set of commands. In addition, the authorization module 124 may also check the value provided to the command deny flag, by an administrator.

[0066] At block 220, the method 200 may include determining, based on the comparison, whether the command is authorized for execution on the authorization system 102. In an implementation, the authorization module 124 may determine the authorization of the command. For example, the authorization module 124 may check availability of the command in the set of commands stored in the cache memory of the authorization system 102. If the command is not available in the set of commands and the value of the command deny flag is OFF for the set of commands, the authorization module 124 may deny the user to execute the command on the authorization system 102.

[0067] At block 222, the method 200 may include authorizing, based on the determination, the user to execute the command on the authorization server 102. In an implementation, the authorization module 124 may be configured to authorize the user.

[0068] Although embodiments for authorization of execution of a command on a remote server have been described in a language specific to structural features or method(s), it is to be understood that the invention is not necessarily limited to the specific features or method(s) described. Rather, the specific features and methods are disclosed as embodiments for authorization of execution of a command on a remote server.

I/We claim:

1. A method for authorization of execution of a command on an authorization system (102), the method comprising:
 - transmitting, by a processor (110), an access request received from a user to a lightweight directory access protocol (LDAP) server (108), wherein the access request includes login credentials of user for authenticating the user to access the authorization system (102);
 - upon authentication, retrieving, based on the login credentials, by the processor (110), a user profile from the LDAP server (108), wherein the user profile includes a set of commands associated with the user, and wherein the set of commands is one of a set of allowed commands and a set of denied commands;
 - storing, by the processor (110), the user profile in a cache memory of the authorization system (102);
 - receiving, by the processor (110), the command to be executed on the authorization system (102);
 - comparing, by the processor (110), the command received from the user with the set of commands stored in the cache memory of the authorization system (102);
 - based on the comparison, determining, by the processor (110), whether the command is authorized for execution on the authorization system (102); and
 - based on the determination, authorizing the user, by the processor (110) to execute the command on the authorization system (102).
2. The method as claimed in claim 1, wherein the authenticating, by the processor (110), comprises establishing a session with the authorization system (102).
3. The method as claimed in claim 2, wherein the session is established through an authentication protocol, wherein the authentication protocol is one of a lightweight directory access protocol (LDAP) session, a terminal access controller access-control system plus (TACACS+) session, a remote authentication dial in user service (RADIUS) session, Local, and Kerberos.

4. The method as claimed in claim 1, wherein the user profile further includes a command deny flag associated with the set of commands, wherein the command deny flag indicates whether the set of commands is executable by the user.
5. The method as claimed in claim 4, wherein the command deny flag is associated with a value, wherein the value is one of ON and OFF.
6. An authorization system (102) for authorization of execution of a command, the authorization system (102) comprising:
 - a processor (110);
 - an authentication module (120), coupled to the processor (110), to,
 - receive an access request from a user, wherein the access request includes login credentials of the user for authenticating the user to access the authorization system (102); and
 - transmit the access request received from a user to a lightweight directory access protocol (LDAP) server (108);
 - a retrieval module (122), coupled to the processor (110), to,
 - retrieve, upon authentication, a user profile from the LDAP server (108), wherein the user profile includes a set of commands associated with the user, and wherein the set of commands is one of a set of allowed commands and a set of denied commands;
 - store the user profile in a cache memory of the authorization system (102);
 - and
 - an authorization module (124), coupled to the processor (110), to,
 - receive, from the user, the command to be executed on the authorization system (102);
 - compare the command received from the user with the set of commands stored in the cache memory of the authorization system (102);
 - determine, based on the comparison, whether the command is authorized for execution on the authorization system (102); and
 - based on the determination, authorize the user to execute the command on the authorization system (102).

7. The authorization system (102) as claimed in claim 6, wherein the authentication module (120) establishes a session with the user.
8. The authorization system (102) as claimed in claim 7, wherein the session is established through an authentication protocol, wherein the authentication protocol is one of a lightweight directory access protocol (LDAP) session, a terminal access controller access-control system plus (TACACS+) session, a remote authentication dial in user service (RADIUS) session, Local, and Kerberos.
9. The authorization system (102) as claimed in claim 6, wherein the user profile further includes a command deny flag associated with the set of commands, wherein the command deny flag indicates whether the set of commands is executable by the user.
10. The authorization system (102) as claimed in claim 9, wherein the command deny flag is associated with a value, wherein the value is one of ON and OFF.
11. A non-transitory computer-readable medium having embodied thereon a computer program for executing a method for determining authorization of execution of a command on an authorization system (102), the method comprising:
 - transmitting, by a processor (110), an access request received from a user to a lightweight directory access protocol (LDAP) server (108), wherein the access request includes login credentials of user for authenticating the user to access the authorization system (102);
 - upon authentication, retrieving, based on the login credentials, by the processor (110), a user profile from the LDAP server (108), wherein the user profile includes a set of commands associated with the user, and wherein the set of commands is one of a set of allowed commands and a set of denied commands;
 - storing, by the processor (110), the user profile in a cache memory of the authorization system (102);
 - receiving, by the processor (110), the command to be executed on the authorization system (102);

comparing, by the processor (110), the command received from the user with the set of commands stored in the cache memory of the authorization system (102);

based on the comparison, determining, by the processor (110), whether the command is authorized for execution on the authorization system (102); and

based on the determination, authorizing the user, by the processor (110) to execute the command on the authorization system (102).

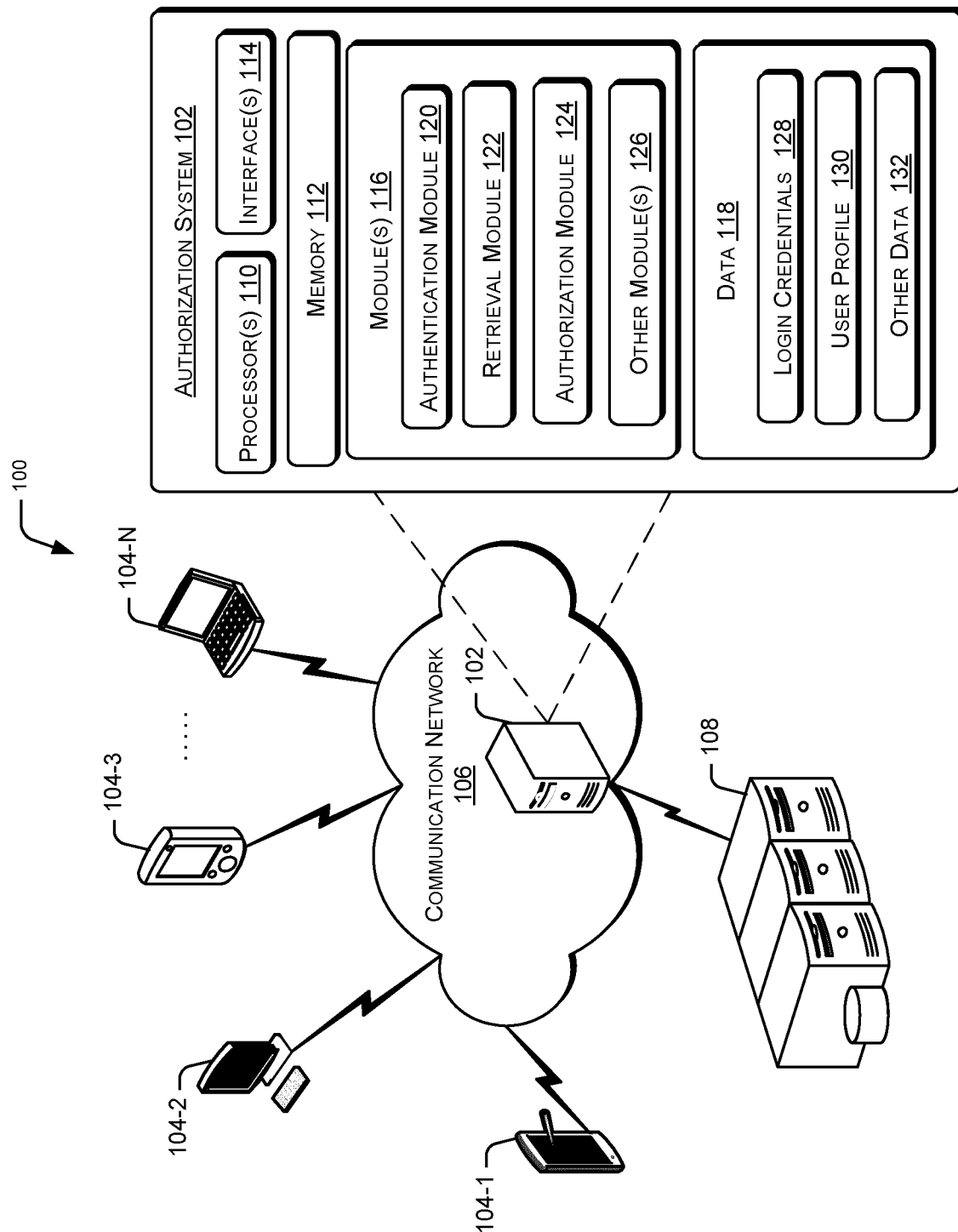


Figure 1

2/2

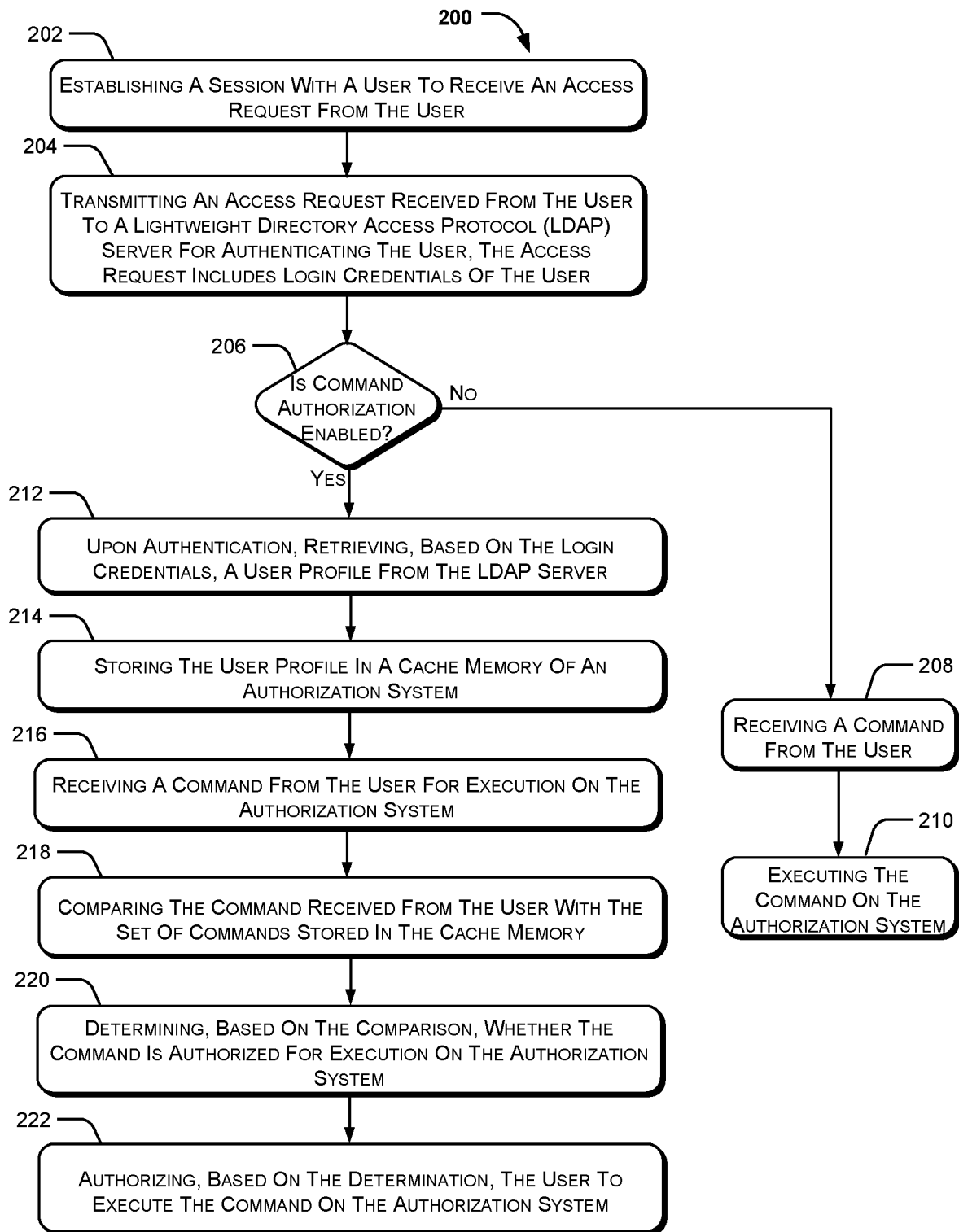


Figure 2

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2014/071228

A. CLASSIFICATION OF SUBJECT MATTER

INV. H04L29/06 G06F21/62 H04L29/08
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data, COMPENDEX, INSPEC, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 7 047 563 B1 (WEBER GREGORY [US] ET AL) 16 May 2006 (2006-05-16) abstract column 1, line 18 - line 44 column 1, line 60 - column 2, line 9 column 2, line 56 - column 4, line 31 figures 1-3	1-11
A	----- WO 02/35797 A2 (NOMADIX INC [US]) 2 May 2002 (2002-05-02) abstract page 10, line 33 - page 11, line 12 page 12, line 9 - line 21 figure 1 ----- -/--	1-11



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

19 December 2014

Date of mailing of the international search report

07/01/2015

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Poppe, Fabrice

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2014/071228

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2008/219241 A1 (LEINONEN ANU [FI] ET AL) 11 September 2008 (2008-09-11) abstract paragraph [0029] paragraph [0116] -----	1-11
A	US 8 205 246 B2 (SHATZKAMER KEVIN [US] ET AL) 19 June 2012 (2012-06-19) abstract column 3, line 45 - column 4, line 18 column 7, line 8 - line 67 figures 1, 2 -----	1-11

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2014/071228

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 7047563	B1	16-05-2006	NONE
WO 0235797	A2	02-05-2002	AU 1336702 A 06-05-2002 WO 0235797 A2 02-05-2002
US 2008219241	A1	11-09-2008	NONE
US 8205246	B2	19-06-2012	US 2008279200 A1 13-11-2008 US 2012227093 A1 06-09-2012