

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
12 September 2002 (12.09.2002)

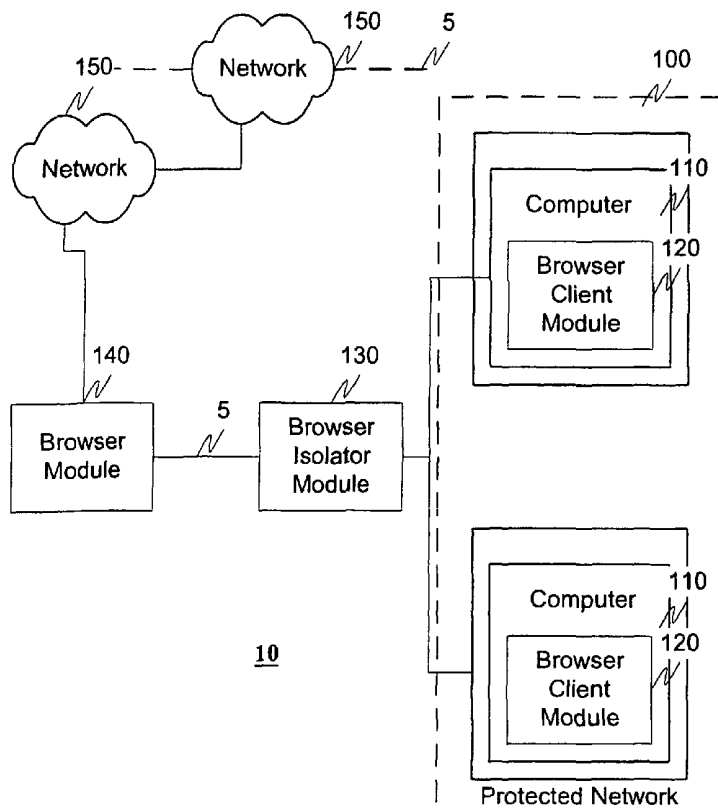
PCT

(10) International Publication Number  
**WO 02/071686 A1**

- (51) International Patent Classification<sup>7</sup>: **H04L 9/12** (74) Agent: **VICK, Jason, H.**; Nixon Peabody LLP, Suite 800, 8180 Greensboro Drive, McLean, VA 22102 (US).
- (21) International Application Number: PCT/US02/06018
- (22) International Filing Date: 1 March 2002 (01.03.2002) (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW.
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:  
60/272,053 1 March 2001 (01.03.2001) US  
60/272,055 1 March 2001 (01.03.2001) US
- (71) Applicant: **INVICTA NETWORKS, INC.** [US/US]; 13873 Park Center Road, Suite 400, Herndon, VA 20171 (US).
- (72) Inventors: **HATFALVI, Emil, J.**; 2700 Thornbrook Road, Ellicott City, MD 21042-7812 (US). **NOBLE, James, G.**; 7802 Lowmoor Ct., Springfield, VA 22153 (US).
- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: SYSTEMS AND METHODS THAT PROVIDE EXTERNAL NETWORK ACCESS FROM A PROTECTED NETWORK



(57) Abstract: A user (110) who is connected to an unprotected network (150) via a protected network (100) is able to browse (120) the Internet without concern that unauthorized code will execute within their local workstation (110) as the result of a vulnerability within the browser executing on a special virtual machine or a browser module (140). Any authorized code will only affect one of a special virtual machine or a browser module (140).

**Published:**

- *with international search report*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

*For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

**SYSTEMS AND METHODS THAT PROVIDE EXTERNAL NETWORK  
ACCESS FROM A PROTECTED NETWORK**

**Related Application Data**

[0001] This application claims the benefit of and priority to U.S. Application Serial No. 60/272,055, entitled "Apparatus and Method to Provide Internet Access For A Protected Network By A Dedicated Browser Computer," filed March 01, 2001, and U.S. Application Serial No. 60/272,053, entitled "Apparatus and Method to Provide Internet Access From A Protected Network By A Tunnelled Communications Path," filed March 01, 2001, both of which are incorporated herein by reference in their entirety.

**Background Of The Invention**

**Field of the Invention**

[0002] The systems and methods of this invention generally relate to network security. Specifically, the systems and methods of this invention provide secure access to an external network or computer from a protected network or computer.

**Description Of Related Art**

[0003] On a private or limited-access protected network, there is a need to provide access to one or more external networks or computers, such as "Internet access," to user workstations for purposes such as browsing, file transfer, etc. (collectively "browsing"). Current browsers run on a user workstation or personal computer and require direct or proxy access to the Internet. This creates a mechanism for the introduction of malicious code onto the user workstation or personal computer.

[0004] Further, malicious code on a user workstation likewise has direct or proxy access to the Internet, raising the possibility of confidential or proprietary information being transmitted to the Internet or external network without the knowledge of the user.

### **Summary of the Invention**

[0005] The exemplary systems and methods of this invention provide external network access, such as Internet access, for a protected network by a dedicated browser. Specifically, an exemplary embodiment of the present invention provides Internet access for users on a private or limited-access/protected network (hereinafter the “protected network”) to the Internet or other network off the protected network by dedicating a browser module(s) to execute the Internet browser and other Internet access programs. The browser module has limited access to the protected network, sufficient to communicate with browser clients on one or more user workstations of the protected network. This provides a mechanism to protect the protected network and the user computer(s) from viruses, worms, trojan horses, and all forms of unauthorized code encountered while accessing the Internet. Furthermore, the browser module prevents unauthorized code on the protected network from communicating with the Internet. The browser module and the browser client module communicate via a restricted protocol. Between the browser module and the protected network is an optional browser isolator module, which can restrict the network communications, such as ethernet communications, to specifically that protocol necessary to support the remote Internet access on the protected computer.

[0006] In accordance with a second exemplary embodiment of this invention, the systems and methods provide external network access, such as Internet access, for users on a private or limited-access/protected network to the Internet or other network off the protected network by running Internet-accessing programs, e.g., browser programs, ftp programs, telnet programs, Voice over IP programs, streaming media programs, or the like, on a special virtual machine on the user workstation. This special virtual machine provides a tunnelled and authenticated communications path from the browser to a border module which then provides access to the Internet. The authenticated tunnel from the special virtual machine to the border module prevents unauthorized code on any workstation on the protected network from accessing the border module and thereby accessing the Internet.

[0007] Accordingly, an exemplary aspect of the systems and methods of this invention is to provide security to a protected network.

[0008] Aspects of the present invention also relate to computer security. In particular, the exemplary embodiments of the systems and methods of this invention relate to providing secure access to an unprotected network.

[0009] Aspects of the present invention also relate to monitoring communication to ensure the communications are performing in an authorized manner.

[0010] Aspects of the present invention also relate to providing a simulated environment via which communication with an unprotected network can be accomplished, thereby protecting the protected environment.

[0011] These and other features and advantages of this invention are described in or are apparent from the following detailed description of the embodiments. For example, the systems and methods of this invention can be used in conjunction with copending U.S. Application Serial No. 09/571,377, entitled "Method of Communications and Communication Network Intrusion Protection Methods And Intrusion Attempt Detection System," U.S. Application Serial No. 09/862,477, entitled "Systems And Methods For Communication Protection," U.S. Application Serial No. 09/867,442, entitled "Systems And Methods For Distributed Network Protection," U.S. Application Serial No. 09/925,503, entitled "Systems And Methods For Distributed Network Protection," and Attorney Docket No. 741946-43, entitled "Systems and Methods For Creating a Code Inspection System," filed February 14, 2002, all of which are incorporated herein by reference in their entirety.

#### **BRIEF DESCRIPTION OF THE DRAWINGS**

[0012] The embodiments of the invention will be described in detail, with reference to the following figures wherein:

[0013] Fig. 1 is a functional block diagram illustrating an exemplary browser module according to this invention;

[0014] Fig. 2 is a functional block diagram illustrating an exemplary tunnelled communications path according to this invention;

[0015] Fig. 3 illustrates an exemplary method of operating the browser interface system according to an embodiment of this invention; and

[0016] Fig. 4 illustrates an exemplary method of establishing a tunnelled communications path according to an exemplary embodiment of this invention.

[0017] Appendix A illustrates exemplary pseudo code for performing connectivity and filtering for communications to/from the protected network to an external network.

### **DETAILED DESCRIPTION OF THE INVENTION**

[0018] Fig. 1 illustrates an exemplary browser interface system 10 that provides Internet access for a protected network through a browser module . Specifically, the browser interface system 10 comprises one or more protected networks 100, one or more protected computers 110, comprising a browser client module 120, a browser isolator module 130, a browser module 140, and one or more networks 150, all connected by links 5.

[0019] While the exemplary embodiments illustrated herein show the various components of the browser interface system and restricted tunneling system collocated, it is to be appreciated that the various components of these systems can be located at distant portions of a distributed network, such as a local area network, a wide area network, a telecommunications network, an intranet and/or the Internet or within a dedicated browser interface system or restricted tunneling system. Thus, it should be appreciated that the components of these systems can be combined into one or more devices or collocated on a particular node of a distributed network, such as a protected network. As will be appreciated from the following description, and for reasons of computational efficiency, the components of these systems can be arranged at any location within a distributed network without affecting the operation of the system.

[0020] Additionally, it is to be appreciated that the term module as referenced herein refers to hardware or software of any combination thereof that is capable of performing the functionality associated with the particular element(s).

[0021] Furthermore, it should be appreciated that the various links 5 connecting the elements can be wired or wireless links, or a combination thereof, or any known or later

developed element(s) that is capable of supplying and/or communicating data to and from the connected elements.

[0022] Additionally, it should be appreciated that while various functional components are illustrated, various other components, such as input and output devices, firewalls, network security devices, or the like, can also be included without affecting the operation of these systems.

[0023] The protected network 100 includes a plurality of protected computers 110, two of which are shown for purposes of illustration. The protected computers 110 in the protected network 100 are connected to a browser module 140 via a browser isolator module 130.

[0024] The browser module 140 is connected to one or more external networks 150 and provides its Internet-access services to the one or more protected computers 110 on the protected network 100. The browser module 140 is separate and physically distinct from the protected computers 110. The protected computers 110 on the protected network 100 each have a browser client module 120 that communicates with the browser module. Any browser-executed code operates not on the protected computer, but rather on the browser module 140. The browser module(s) is/are sacrificial, in that if unauthorized code is encountered and executed in the process of Internet access, only the browser module 140 is corrupted, and that module can be, for example, taken off-line until repaired. Thus the protected computer 110 and data/programs thereon are not corrupted.

[0025] Optionally, another browser module (not shown) may serve in the absence of the corrupted browser module. The browser module 140 may have software/hardware installed to detect anomalies caused by unauthorized code and may optionally shut itself down automatically. The browser module 140 can also have logically separate access to both the external network 150 and the protected network 100. The browser module 140 protocol-level access to the protected network can also be limited to that specifically necessary to support authorized protected computer 110 access.

[0026] The protected computer's 110 access to the protected network is limited to those communications specifically necessary for the remote operation of the browser module 140 and the user feedback (audio and visual) of the same. Other communications are generally prohibited. This isolation can be accomplished in hardware and/or software. For example,

the protocol can involve a TCP connection initiated by the protected computer to the browser module 140 acting as a "server." The browser module 140 could then not permit TCP connections initiated by an unprotected computer directed towards a protected computer 110 on the protected network. Likewise, the specific TCP ports which can be addressed in either direction could be limited to those required by the browser client module 120. In this manner, damage to computers on the protected network is further prevented in the case of a compromised browser module 140. This function can be accomplished in software and/or hardware on the protected computer 110, or, for greater security, in hardware between the protected computer and the protected network. The hardware configuration has the advantage of not being corruptible by unauthorized code inadvertently loaded onto the browser computer. The hardware device can also have a separate network connection (not shown) for each side (the browser computer and the protected network) and could act as a restrictive router between the two.

**[0027]** The browser client module 120 on the protected computer 110 provides the user with the control of and the visual and/or audio output of the actual browser operating remotely on the browser module 140. This operates analogously to robotic manipulators used in handling dangerous materials, e.g., explosives, except that the browser is operated remotely on a sacrificial browser computer, e.g., the browser module 140, and the audio and video appears as if it would to a local browser on the protected computer 110.

**[0028]** Additionally, the user(s) may download files from the Internet just as they normally would from a local browser. However, in this case, the files are initially downloaded onto the browser module 140.

**[0029]** The browser module 140 and browser isolator module 130 prevent unauthorized access to the protected network 100. The browser isolator module 130 prevents communications between the browser module 140 and the protected network 100 other than those particular authorized messages necessary for the remote operation of the protected computer 110. Thus, the ability of a compromised browser module 140 to adversely affect the protected computers 110 on the protected network 100 is minimized.

**[0030]** Unauthorized code brought onto the protected network via another mechanism, e.g., a user's floppy disk, is prevented from direct access to the Internet, since only the

browser client module 120 on the protected computer 110 can communicate with the browser module 140. The browser isolator module 130 is also capable of restricting this communication. Similarly, individual users are prevented from direct access to the Internet, except as provided by the browser client module 120 and the protected computer 110. The browser client module 120 and the protected computer 110 may also both monitor the characteristics of the user's browser communications with the browser module 140 to validate that it is indeed not being manipulated by, for example, a user-computer-resident unauthorized code. The browser module may also periodically validate that an actual person is controlling the browser client module 120.

**[0031]** One exemplary implementation of the systems and methods of this invention uses two network protocols from the X-Windows system: The "X Window System Protocol" (the X protocol) by the X Consortium Standard, X Version 11, Release 6.4, and the "X Display Manager Control Protocol" (the XDMCP protocol), by the X Consortium Standard X Version 11, Release 6.4, both of which are incorporated herein by reference in their entirety. However, it is to be appreciated that any protocol/system/software can be used with equal success that allows a first computer to execute and control applications that run on a second computer.

**[0032]** These are both client/server protocols and are used together to allow a user on one computer to execute and control applications that run on another computer. In the context of Figure 1, the user is working at one of the protected computers 110 and executes and controls applications that run on the browser module 140 by means of the X and XDMCP protocols. The applications run by the user on the protected computer 110 can be any applications supported by that computer system, including those that access the Internet. The browser isolator module 130 ensures that communication with the protected computer 110 is restricted to these two protocols.

**[0033]** In order for a user working at one of the protected computers 110 to execute applications on the browser module 140, an X-Windows session is established between the two computers. The first step towards this goal is accomplished by the XDMCP protocol. For this protocol, the protected computer 110 is the client and the browser module 140 is the server. The protected computer 110 starts by sending an XDMCP request to the browser module 140 requesting that it establish an X-Windows session with the protected computer.

The browser module 140 responds by indicating its willingness to establish an X-Windows session and, assuming it is willing, several more XDMCP requests and responses are exchanged. In a successful scenario this results in the browser module 140 initiating an X-Windows session with the protected computer 110.

**[0034]** When the browser module 140 establishes an X-Windows session with the protected computer 110, the user can be authenticated through the browser module's login process, e.g. providing a username and password, to gain access to the browser module 140. Once authentication is validated, the user will be able to execute applications on the browser module 140.

**[0035]** What has been described thus far is just the normal operation of the XDMCP and X protocols. One focus of this invention is the operation of the browser isolator module 120, which scrutinizes the network communications between the protected computer 110 and the browser module 140. The selected protocols of the X-Windows system have the attribute that they do not transfer data between computers other than that necessary for the user interface. The user interface data is temporary in that it is transferred across the network and used to generate the interface shown to the user, but is not stored permanently. This is unlike the Internet File Transfer Protocol (FTP), which copies the contents of files from one computer to another and permanently stores the copies on the receiving computer. By restricting communication with the protected computer 110 to these two X-Windows protocols, the browser isolator module 120 prevents the transfer of permanent data between one or more of the protected computers 110 and the browser module 140, and from there the external network 150.

**[0036]** If the browser isolator module 120 only restricted communications with the browser module 140 to certain protocols, the browser isolator module 120 would be nothing more than a simple firewall. However, the browser isolator module 120 not only restricts communication to the XDMCP and X protocols, it can examine each protocol packet in detail, inspecting one or more predetermined fields in the packets to ensure that each field contains valid, e.g., authorized, information. Beyond verifying that each field contains valid information, fields with related information can be compared within and between packets in the protocol session to ensure that there is a consistent relationship during the life of the

session. Individual field contents can also be verified in a variety of ways depending on the type of the field.

**[0037]** For example, text fields can be checked against a list of expected values where known. If a text field is not restricted to a list of known values, it may still have a predictable pattern to which it should conform, e.g., a capital letter followed by three decimal digits. If so, this can be checked and enforced. At a minimum, text fields can be checked to ensure that they are composed of, for example, printable characters. In general, any information can be checked to insure the information is what is expected to for particular communication and/or environment.

**[0038]** For example, several XDMCP packets can contain a field with one or more Authorization Names. Field values can be checked against a list of known values that consists of the strings "MIT-KERBEROS-4", "MIT-KERBEROS-5", "MIT-MAGIC-COOKIE-1", "SUN-DES-1", "XC-QUERY-SECURITY-1", and "XDM-AUTHORIZATION-1." Any Authorization Name that does not match one of the values in this list can be rejected.

**[0039]** Numeric fields either can have a restricted set of values or can be unconstrained. Fields with a restricted set of values can be checked against the restricted set. Some numeric fields can have unconstrained values but a fixed length that can also be checked.

**[0040]** The value of one field can restrict the permitted values or length of another field. These relationships can be enforced both within a packet and between packets. Some fields are expected to have the same value over a long period of time, even spanning system restarts.

**[0041]** The exemplary purpose of these detailed packet field checks by the browser isolator module 120 is two-fold. By ensuring that the fields contain only valid, consistent information, the browser isolator module 120 reduces the chances that a defect in the protocol implementation on either the browser module 140 or a protected computer 110 can be exploited by unauthorized software. The checks for data consistency also reduce the possibility that packet fields can be used to surreptitiously transfer data by varying their contents over time.

**[0042]** For example, during the XDMCP packet exchange the protected computer can specify an X-Windows display number, which has legal values between 0 and 63. When the browser module initiates the X-Window session with the protected computer it does so on a TCP port number that is derived from the display number (display number plus 6000). The display number specified in the XDMCP packets can be stored in a profile for that user's workstation. When the browser module starts the X-Windows session can must use the TCP port that corresponds to the display number or the connection can be rejected.

**[0043]** Appendix A provides a high-level description of the exemplary processing logic that can be applied to one or more packets sent to or sent from the browser module 140. For example, the following information can be screened to determine if the communication is authorized: source and destination ports, user information, direction of information, host information, destination information, character information, IP address information, display identification, session information, display class, display number, TCP information, date and/or time information, or the like. In general, an information communicated between the browser and the browser client can be used. Furthermore, there are a plurality of options that can be invoked upon determining that one or more packets are authorized including deleting the packet, further analyzing the packet, determining the origin of the packet or the like.

**[0044]** Alternatively, or in conjunction with the above, the systems and methods of this invention can provide Internet access from a protected network 200 by a tunnelled communications path. As illustrated in Fig. 2, the protected network 200 comprises one or more protected computers 210, two of which are shown for purposes of illustration. Each protected computer 210 comprises a special virtual machine (SVM) 220 having a browser 230. The browser 230 for each protected computer 120 communicates with the external network 150, such as the Internet by means of a border module 240.

**[0045]** The browser 230 can be any browser which runs on a virtual machine or an emulated machine. For example, the browser 230 could be a browser written in Java® running on a Java® Virtual Machine or a Windows®-based browser running in a "virtual PC," or, in general, any browser not running in a "native mode" on the user workstation.

**[0046]** The special virtual machine 220 can be a traditional virtual machine (VM), such as a Java® VM, which has had its communications ability modified so as to tunnel and

authenticate all network communications to the border module 240. The normal network communications ability of the VM would traditionally allow an application direct local network and Internet access in a straightforward manner, for example by creating sockets, sending and receiving packets, etc. The communications protocol stacks of the SVM 220 are modified so that all communications are tunnelled and authenticated to the border module 240 for transmission to/from the external network 150. This tunnelling and authentication is transparent to the browser 230 and any other applications running on the SVM. Any application running on the virtual machine, such as a browser, Internet Relay Chat, e-mail, streaming media, or the like, would appear to have direct access to the external network. Applications running on the SVM would have no access to the protected network or its traffic, since all network communications paths are tunnelled to the border module 240. Applications on the protected computer 210 other than the SVM 220 would not have access to the external network since they would lack access to the authenticated tunnel.

[0047] The border module 240 is a component which has at least two logical and/or physical network connections: one to the protected network 200 and another to the external network 150. The border module 240 encapsulates and tunnels inbound external network communications to the end-user workstations' SVMs on the protected network 200. Outbound traffic from the SVMs 220 is authenticated, de-capsulated, and forwarded to the external network 150 as normal traffic. The authentication of the communications between the SVM 220 and the border module 240 prevents unauthorized or unauthorized native code anywhere on the protected network from accessing the external network 150, since only the SVM 220 has the authentication necessary to accomplish the transfer of the restricted data. Generally, unauthorized code on the SVM 220 is not prevented from accessing the external network 150. However, if the SVM 220 acquires unauthorized code, such as a virus or trojan horse, the unauthorized code would not have access to the information on the protected network 200, and the SVM 220 could simply be re-loaded, reset, or the like, for example, automatically or by a user, thus removing the unauthorized code.

[0048] To further enhance the security of the protected network 200, an optional tunnel restrictor module 250 may be placed between the border module 240 and the protected network 200. The tunnel restrictor module 250 can limit the border module's access to the protected network 200 to those communications specifically necessary for the tunnelled

communications with the SVM's on the protected computers 210 on the protected network 200.

[0049] Other communications can be prevented bi-directionally by the tunnel restrictor module 250. This isolation can be accomplished in hardware and/or software, by a unit placed between the border module and the protected network. This unit could have at least two network interfaces: one for the protected network side, and one for the border processor side. For example, the tunnel protocol could involve a TCP connection initiated by the SVM 220 on a user workstation to the border module 240 acting as the "server." The tunnel restrictor module 250 could then not permit TCP connections initiated by the border module 240 directed towards a user workstation 210 on the protected network 200. Likewise, the specific TCP or UDP ports which can be addressed in either direction would be limited to those required by the SVM. In this manner, damage to computers on the protected network is further prevented in case of a compromised border module 240. Likewise, external e-mail may also be sent or received via the browser 230 running on the SVM 220.

[0050] The border module 240 tunnels and thereby encapsulates outside traffic as it travels on the protected network 200 to the special virtual machines 220. Therefore, outside access to the protected computer 210 and to the normal traffic of the protected network 200 is prevented. The tunnel restrictor module 250 further prevents access to the protected network 200 from the external network 150 should the border processor module 240 be compromised.

[0051] Unauthorized code brought onto the protected network 200 via another mechanism, such as a user's floppy disk, is prevented from direct access to the Internet, since only the special virtual machine 220 on the protected computer 210 can communicate with the border module 240, and thereby the external network. The optional tunnel restrictor module 250 can also prevent this type of communication. Similarly, individual users are prevented from direct access to the Internet, except as provided by the browser 230 or other software on the special virtual machine 220.

[0052] Fig. 3 illustrates an exemplary method of operating the browser interface system according to this invention. In particular, control begins in step S100 and continues to step S110. In step S110, communications are established between the browser and the browser client. Next, in step S120, communications between the browser and the browser client are

inspected. Then, in step S130, a determination is made whether the communications are authorized.

[0053] If the communications are authorized, control jumps to step S150. Otherwise, control continues to step S140. In step S140, the unauthorized information is processed in accordance with a predetermined procedure. Control then continues to step S160.

[0054] In step S150, the authorized communications are allowed. Control then continues to step S160.

[0055] In step S160, a determination is made whether to end the communication session. If the session is complete, control continues to step S160 where the control sequence ends. Otherwise, control jumps back to step S130.

[0056] Fig. 4 outlines an exemplary method of establishing and monitoring a tunnelled communications path according to an exemplary embodiment of this invention. Specifically, control begins in step S200 and continues to step S210. In step S210, the browser of the virtual machine is enabled. Next, in step S220, communications between the browser of the virtual machine and the border module are established. Then, in step S230, the communications to/from the virtual machine are authenticated. Control then continues to step S240.

[0057] In step S240, a determination is made whether the communications are authorized. If the communications are authorized, control jumps to step S260 where the communications are allowed. Otherwise, control continues to step S250.

[0058] In step S250, the unauthorized communications are processed accordingly. Control then continues to step S270 where the control sequence ends.

[0059] As shown in Figs. 1-2, the browser interface system and restricted tunneling system can be implemented either on a single programmed general purpose computer or a separate programmed general purpose computer. However, the browser interface system and restricted tunneling system can also be implemented on a special purpose computer, a programmed microprocessor or microcontroller and peripheral integrated circuit element, an

ASIC or other integrated circuit, a digital signal processor, a hardwired electronic or logic circuit such as a discrete element circuit, a programmable logic device such as PLD, PLA, FPGA, PAL, or the like. In general, any device capable of implementing a finite state machine that is in turn capable of implementing the attached pseudo-code and performing the steps in the accompanying flowcharts can be used to implement the browser interface system and restricted tunneling system according to this invention.

**[0060]** Furthermore, the disclosed method may be readily implemented in software using object or object-oriented software development environments that provide portable source code that can be used on a variety of computer or workstation hardware platforms.

Alternatively, the disclosed browser interface system and restricted tunneling system can be implemented partially or fully in hardware using standard logic circuits or VLSI design.

Whether software or hardware is used to implement the systems in accordance with this invention is dependent on the speed and/or efficiency requirements of the system, the particular function, and the particular software and/or hardware systems or microprocessor or microcomputer systems being utilized. However, the browser interface system and restricted tunneling system and methods illustrated herein can be readily implemented in hardware and/or software using any know or later-developed systems or structures, devices and/or software by those of ordinary skill in the applicable art from the functional description provided herein and with a general basic knowledge of the computer arts.

**[0061]** Moreover, the disclosed methods may be readily implemented as software executed on a programmed general purpose computer, a special purpose computer, a microprocessor, or the like. In these instances, the methods and systems of this invention can be implemented as a program embedded on a personal computer, such as a JAVA® or CGI script, as a resource residing on a server or workstation, a routine embedded on a dedicated browser interface system and restricted tunneling system, a web browser, a PDA, a dedicated browser interface system, a dedicated restricted tunneling system, or the like. The browser interface system and restricted tunneling system can also be implemented by physically incorporating the system into a software and/or hardware system, such as the hardware and software systems of a computer workstation or dedicated browser interface system and restricted tunneling system.

[0062] It is, therefore, apparent there has been provided in accordance with the present invention, systems and methods for protecting a protected network. While this invention has been described in conjunction with a number of embodiments, it is evident that many alternatives, modifications, and variations would be or are apparent those of ordinary skill in the applicable art. Accordingly, the invention is intended to embrace all such alternatives, modifications, equivalents and variations that are within the spirit and scope of this invention.

APPENDIX A

```

IF (it is time to perform housekeeping activities) THEN
5   FOR (each TCP connection that has started to close, i.e. observed the TCP FIN flag set) DO
      IF (the time allowed for closure has passed) THEN
          Close this connection
      IF (this is the last connection with this user) THEN
          Reinitialize the user's profile and set the profile state to Incomplete
          ENDF
      ENDF
      NEXT
10
15  FOR (each connection that has been idle for the maximum allowed time) DO
      Close this connection
      IF (this is the last connection with this user) THEN
          Reinitialize the user's profile and set the profile state to Incomplete
          ENDF
          NEXT
20  ENDF
      ENDF
      NEXT
      ENDF
25  IF (it is an IP packet) THEN
      IF (it is an IP/UDP packet) THEN
          Check the source and destination ports against a list of allowed well-known UDP ports
      IF (either port is on the allowed port list) THEN
          IF (the server port is for the XDMCP protocol) THEN
              Check the XDMCP Opcode against a list of allowed opcodes
          IF (the XDMCP Opcode is on the allowed opcode list) THEN
30

```

```
35      Check if the XDMCP packet is being sent in an allowed direction
      IF (an allowed Opcode is being sent to the server or an allowed Opcode is being sent to the user) THEN
      IF (there is an established X-Windows connection with this user) THEN
      IF (the Opcode is other than KeepAlive or Alive) THEN
      Discard the packet
      ENDIF
40      ELSE
      IF (the Opcode is KeepAlive or Alive) THEN
      Discard the packet
      ENDIF
      ENDIF
45      Create and initialize a profile for this user
      Set the profile state to Incomplete
17      IF (the Opcode is Query) THEN
50      Reinitialize the user's profile and set the profile state to Incomplete
      FOR (each Authentication Name in the packet) DO
      IF (the Authentication Name is not recognized) THEN
      Discard the packet
      ENDIF
55      NEXT
      Remember the Authentication Names in the user's profile
60      ELSE IF (the Opcode is Willing) THEN
      IF (the Authentication Name in the packet is not recognized) THEN
      Discard the packet
      ENDIF
```

65 IF (the Authentication Name is not one of the authentication names recorded in the user's  
profile from the Query packet) THEN  
Discard the packet  
ENDIF

70 Remember this Authentication Name in the user's profile

IF (the Hostname supplied by the server is NOT syntactically valid) THEN  
Discard the packet  
ENDIF

IF (the Status string is NOT composed of printable characters) THEN  
Discard the packet  
ENDIF

80 ELSE IF (the Opcode is Unwilling) THEN  
IF (the Hostname supplied by the server is NOT syntactically valid) THEN  
Discard the packet  
ENDIF

85 IF (the Status string is NOT composed of printable characters) THEN  
Discard the packet  
ENDIF

90 Reinitialize the user's profile and set the profile state to Incomplete

ELSE IF (the Opcode is Request) THEN  
IF (the X Display Number is invalid) THEN  
Discard the packet  
ENDIF

95

Remember this Display Number in the user's profile

```
FOR (each Connection Type in the packet) DO
  IF (the Connection Type is not IP) THEN
    Discard the packet
  ENDIF
NEXT
```

```
FOR (each Connection Address in the packet) DO
  IF (the Connection Address is not an IP address) THEN
    Discard the packet
  ENDIF
```

```
IF (the Connection Address does not match the user's IP address) THEN
  Replace the Connection Address with the user's IP address
ENDIF
NEXT
```

```
IF (none of the Connection Addresses matched the user's IP address) THEN
  Discard the packet
ENDIF
```

```
IF (the Authentication Name in the packet is not recognized) THEN
  Discard the packet
ENDIF
```

```
IF (the Authentication Name does not match the Authentication Name recorded in the user's
profile from the Willing packet) THEN
  Discard the packet
ENDIF
```

100

105

110

19

115

120

125

130 IF (the lengths of the Authentication Name and Authentication Data are not both zero or  
non-zero) THEN  
Discard the packet  
ENDIF

135 FOR (each Authorization Name in the packet) DO  
IF (the Authorization Name is not recognized) THEN  
Discard the packet  
ENDIF  
NEXT

140 Remember the Authorization Names in the user's profile

IF (if the Manufacturer Display ID contains invalid characters) THEN  
Discard the packet  
ENDIF

145 IF (if the Manufacturer Display ID does not match any previous recorded Display ID for this  
user) THEN  
Discard the packet  
ENDIF

ELSE IF (the opcode is Accept) THEN  
Remember the Session ID in the user's profile

150 IF (the Authentication Name in the packet is not recognized) THEN  
Discard the packet  
ENDIF

155 IF (the Authentication Name does not match the Authentication Name recorded in the user's  
profile from the Request packet) THEN  
Discard the packet

160                   ENDIF

          IF (the lengths of the Authentication Name and Authentication Data are not both zero or  
          non-zero) THEN  
              Discard the packet  
165           ENDIF

          IF (the Authorization Name in the packet is not recognized) THEN  
              Discard the packet  
170           ENDIF

          IF (the Authorization Name does not match any of the Authorization Names recorded in the  
          user's profile from the Request packet) THEN  
              Discard the packet  
          ENDIF

          Remember the Authorization Name in the user's profile

175           IF (the lengths of the Authorization Name and Authorization Data are not both zero or non-  
          zero) THEN  
              Discard the packet  
180           ENDIF

          ELSE IF (the opcode is Decline) THEN  
              IF (the Status string is NOT composed of printable characters) THEN  
                  Discard the packet  
185           ENDIF

          IF (the Authentication Name in the packet is not recognized) THEN  
              Discard the packet  
190           ENDIF

195 IF (the Authentication Name does not match the Authentication Name recorded in the user's  
profile from the Request packet) THEN  
Discard the packet  
ENDIF

200 IF (the lengths of the Authentication Name and Authentication Data are not both zero or  
non-zero) THEN  
Discard the packet  
ENDIF

Reinitialize the user's profile and set the profile state to Incomplete

205 ELSE IF (the opcode is Manage) THEN  
IF (the Session ID does not match the Session ID recorded in the user's profile from the  
Accept packet) THEN  
Discard the packet  
ENDIF

210 IF (the X Display Number is invalid) THEN  
Discard the packet  
ENDIF

215 IF (the Display Number does not match the Display Number recorded in the user's profile  
from the Request packet) THEN  
Discard the packet  
ENDIF

220 IF (if the Display Class contains invalid characters) THEN  
Discard the packet  
ENDIF

```
225 IF (if the Display Class does not match any previous recorded Display Class for this user)
    THEN
        Discard the packet
    ENDIF
    Set the user's profile state to Complete
230 ELSE IF (the opcode is Refuse) THEN
    IF (the Session ID does not match the Session ID recorded in the user's profile from the
        Accept packet) THEN
        Discard the packet
    ENDIF
235 ELSE IF (the opcode is Failed) THEN
    IF (the Session ID does not match the Session ID recorded in the user's profile from the
        Accept packet) THEN
        Discard the packet
    ENDIF
240 IF (the Status string is NOT composed of printable characters) THEN
        Discard the packet
    ENDIF
245 Reinitialize the user's profile and set the profile state to Incomplete
    ELSE IF (the opcode is KeepAlive) THEN
        IF (the X Display Number is invalid) THEN
            Discard the packet
        ENDIF
250 IF (the Display Number does not match the Display Number recorded in the user's profile
    from the Manage packet) THEN
```

```
255         Discard the packet
        ENDIF

        IF (the Session ID does not match the Session ID recorded in the user's profile from the
        Manage packet) THEN
260             Discard the packet
        ENDIF

        ELSE IF (the opcode is Alive) THEN
            IF (the Session Running flag is not 0 or 1) THEN
265                 Discard the packet
            ENDIF

            IF (the Session Running flag is True) THEN
                IF (the Session ID does not match the Session ID recorded in the user's profile from
                the Manage packet) THEN
270                     Discard the packet
                ENDIF
                ELSE – the Session Running flag is False
                IF (the Session ID is not zero) THEN
275                     Discard the packet
                ENDIF

                Reinitialize the user's profile and set the profile state to Incomplete
                ENDIF

                ELSE – the XDMCP Opcode is not allowed
                Discard the packet
                ENDIF
                ELSE – the XDMCP Opcode is being sent in the wrong direction
280                 Discard the packet
                ENDIF
```

```
290     ELSE -- the XDMCP Opcode is not on the allowed-opcodes list
        Discard the packet
        ENDIF
        ELSE -- the server port is not for the XDMCP protocol
        Discard the packet
        ENDIF
        ELSE -- neither of the IP ports are on the allowed-ports list
        Discard the packet
        ENDIF
295
        ELSE IF (it is an IP/TCP packet) THEN
            IF (this packet is for an established TCP connection) THEN
                IF (the TCP RST flag is set) THEN
300                    Close this connection immediately
                ELSE
                    IF (this is the last connection with this user) THEN
                        Reinitialize the user's profile and set the profile state to Incomplete
                        ENDIF
305
                    ELSE IF (the TCP FIN flag is set for the first time for this connection) THEN
                        Remember that the FIN flag was set in the user's profile
                        Record that the FIN flag was observed at the current date/time
                        ENDIF
310
                    ELSE -- the packet is not for an established TCP connection
                        IF (the packet TCP SYN flag is set and the TCP ACK flag is not set) THEN
                            Check the source and destination ports against a list of allowed well-known TCP ports
                        ELSE IF (either port is on the allowed port list) THEN
                            Check if the TCP connection is being opened in an allowed direction, i.e. the SYN packet is from the
                            server to an allowed user port or from the user to an allowed server port
315
```

```
320 IF (the TCP connection is being opened in an allowed direction) THEN
    Check if the user port is for the X-Windows protocol
    IF (this is an X-Windows connection) THEN
        IF (there is not an existing profile for this user) THEN
            Discard the packet
        ELSE – there is an existing profile for this user
            IF (the state of the user's profile is not Complete) THEN
                Discard the packet
            ENDIF
        ENDIF
    ENDIF
    IF (the user port does not correspond to the X Display Number in the user's profile)
    THEN
        Discard the packet
    ENDIF
    ENDIF
    ENDIF
    Remember this connection in the user's profile
    ELSE – the TCP connection is being opened in a disallowed direction
        Discard the packet
    ENDIF
    ELSE – neither of the IP ports are on the allowed-ports list
        Discard the packet
    ENDIF
    ELSE – this is not a TCP SYN packet
        Discard the packet
    ENDIF
    ENDIF
    Remember the current date/time as that of the last packet on this connection
    ELSE – it is neither and IP/UDP or IP/TCP packet
```

```
        Discard the packet
    ENDIF
ELSE -- it is a non-IP packet
    Discard the packet
355  ENDIF
```

We claim:

1. A communications isolation system comprising:  
a browser module that provides communications access to an unprotected network from a protected network;  
a browser client module that communicates with the browser module; and  
a browser isolator module that analyzes communications between the browser module and the browser client module, wherein the communication between the browser module and the browser client module are limited to those communications necessary for remote operation of the browser module.
2. The system of claim 1, further comprising a browser isolator module that is capable of preventing unauthorized communications between the browser module and the browser client module.
3. The system of claim 2, wherein the browser isolator module screens at least one of the following types of information to determine if the communication is authorized: source and destination ports, user information, origination information, host information, destination information, character information, IP address information, display identification, session information, display class, display number, TCP information, and date and/or time information.
4. The system of claim 1, wherein the browser module comprises a distributed network browser.
5. The system of claim 1, wherein the protected network is isolated from unauthorized communications received from the unprotected network.
6. A communications isolation system comprising:  
a browser running on a virtual machine within a protected network;  
a border module that tunnels communications from an unprotected network to the browser via a communications tunnel, wherein

only authorized communications are allowed between the browser and the unprotected network.

7. The system of claim 6, further comprising a tunnel restrictor module that limits communications from the border module to the virtual machine to those communications authorized for browser operability.

8. The system of claim 6, wherein the virtual machine is isolated from the protected network.

9. The system of claim 6, wherein communications from the unprotected network are encapsulated and forwarded to the virtual machine.

10. The system of claim 6, wherein protected information on the protected network is prevented from reaching the unprotected network.

11. A method of communicating with an unprotected network comprising:  
establishing communications between a browser and a browser client;  
inspecting the communications between the browser and the browser client;  
determining if the communications are authorized; and  
allowing the authorized communications between the browser and the browser client.

12. The method of claim 11, wherein the communication between the browser module and the browser client module are limited to those communications necessary for remote operation of the browser module.

13. The method of claim 11, further comprising screening at least one of the following types of information to determine if the communications are authorized: source and destination ports, user information, origination information, host information, destination information, character information, IP address information, display identification, session information, display class, display number, TCP information, and date and/or time information.

14. A method of establishing a restricted communications tunnel comprising:  
enabling a browser on a virtual machine that is isolated from a protected network;  
establishing communications with a border module;  
tunneling communications from the border module to the browser; and  
preventing unauthorized communications from reaching the protected network.
15. The method of claim 14, further comprising limiting communications from the border module to the virtual machine to those communications authorized for browser operability.
16. The method of claim 14, wherein communications from the unprotected network are encapsulated and forwarded to the virtual machine.
17. An information storage media comprising information for communicating with an unprotected network comprising:  
information that establishes communications between a browser and a browser client;  
information that inspects the communications between the browser and the browser client;  
information that determines if the communications are authorized; and  
information that allows the authorized communications between the browser and the browser client.
18. An information storage media comprising information for establishing a restricted communications tunnel comprising:  
information that enables a browser on a virtual machine that is isolated from a protected network;  
information that establishes communications with a border module;  
information that tunnels communications from the border module to the browser; and  
information that prevents unauthorized communications from reaching the protected network.

19. A communications isolation system comprising:  
means for providing communications access to an unprotected network from a protected network;  
means for communicating with a browser module; and  
means for analyzing communications between the browser module and the a browser client module, wherein the communication between the browser module and the browser client module are limited to those communications necessary for remote operation of the browser module.

20. A communications isolation system comprising:  
means for running a virtual machine within a protected network; and  
means for tunnelling communications from an unprotected network to a browser running on the virtual machine via a communications tunnel, wherein  
only authorized communications are allowed between the browser and the unprotected network.

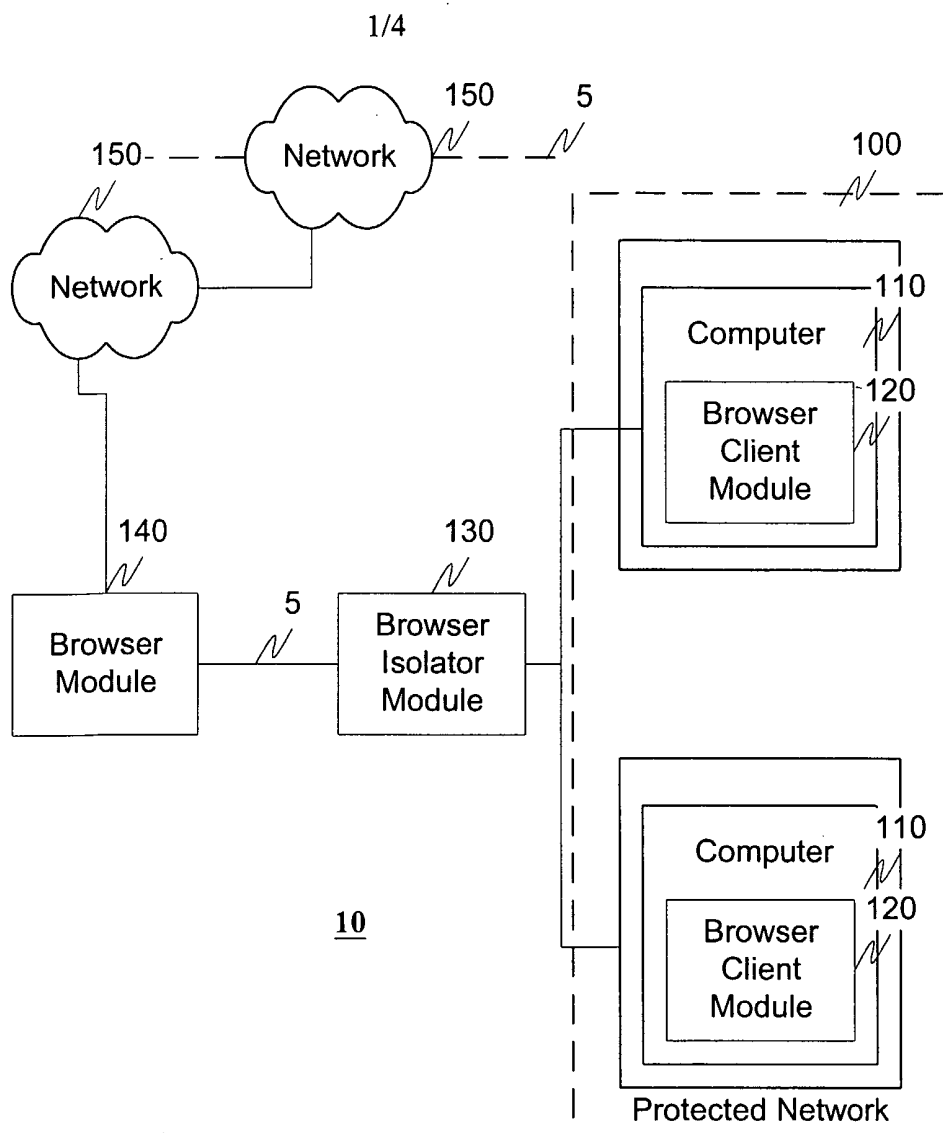


Fig. 1

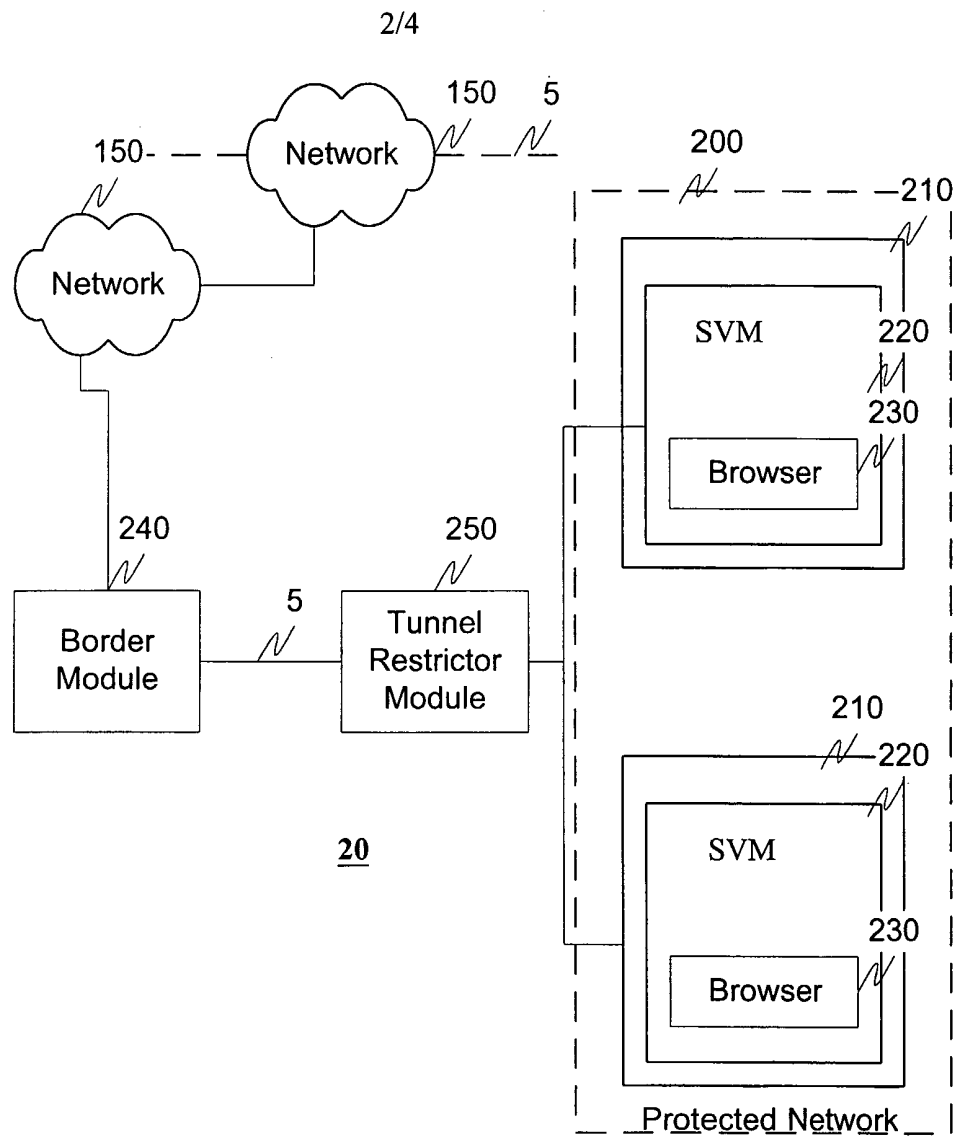


Fig. 2

3/4

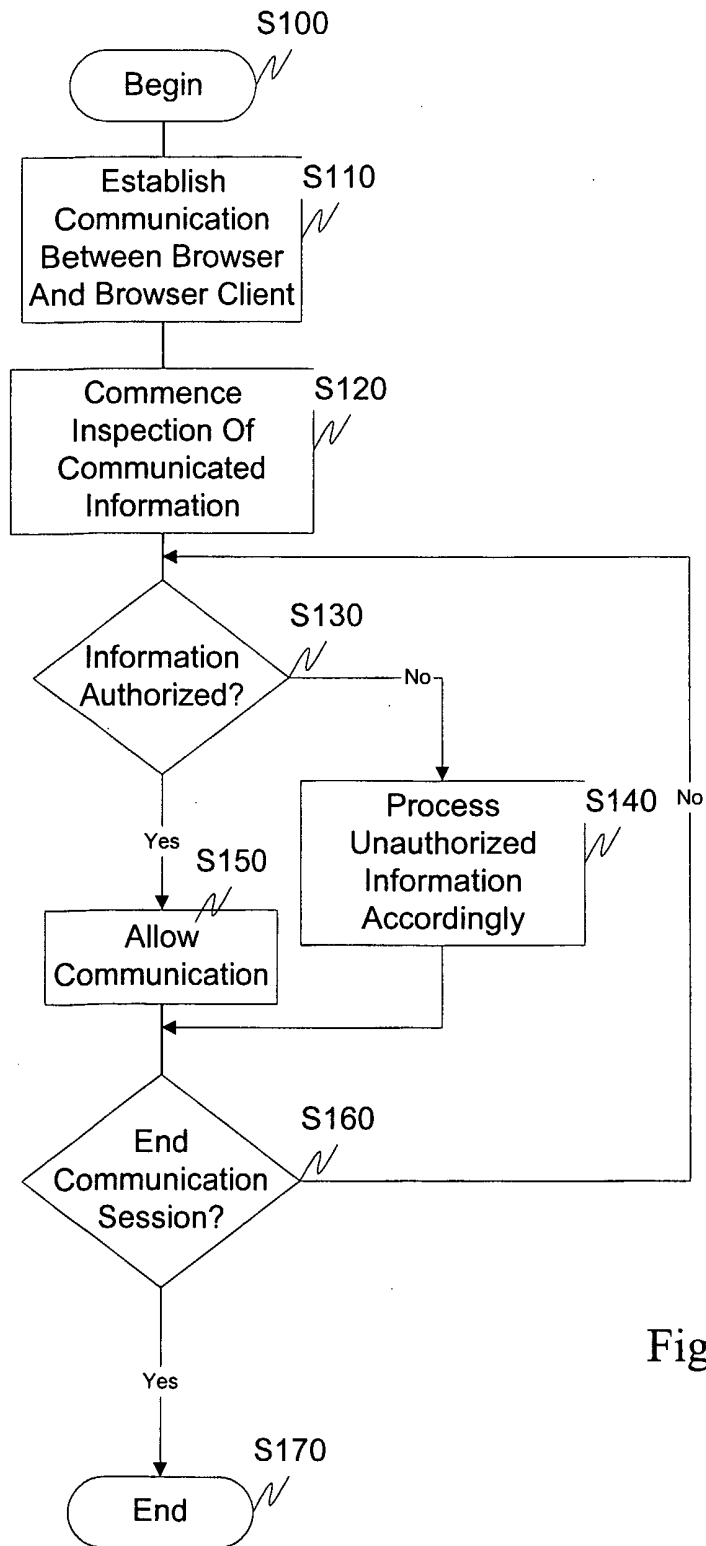


Fig. 3

4/4

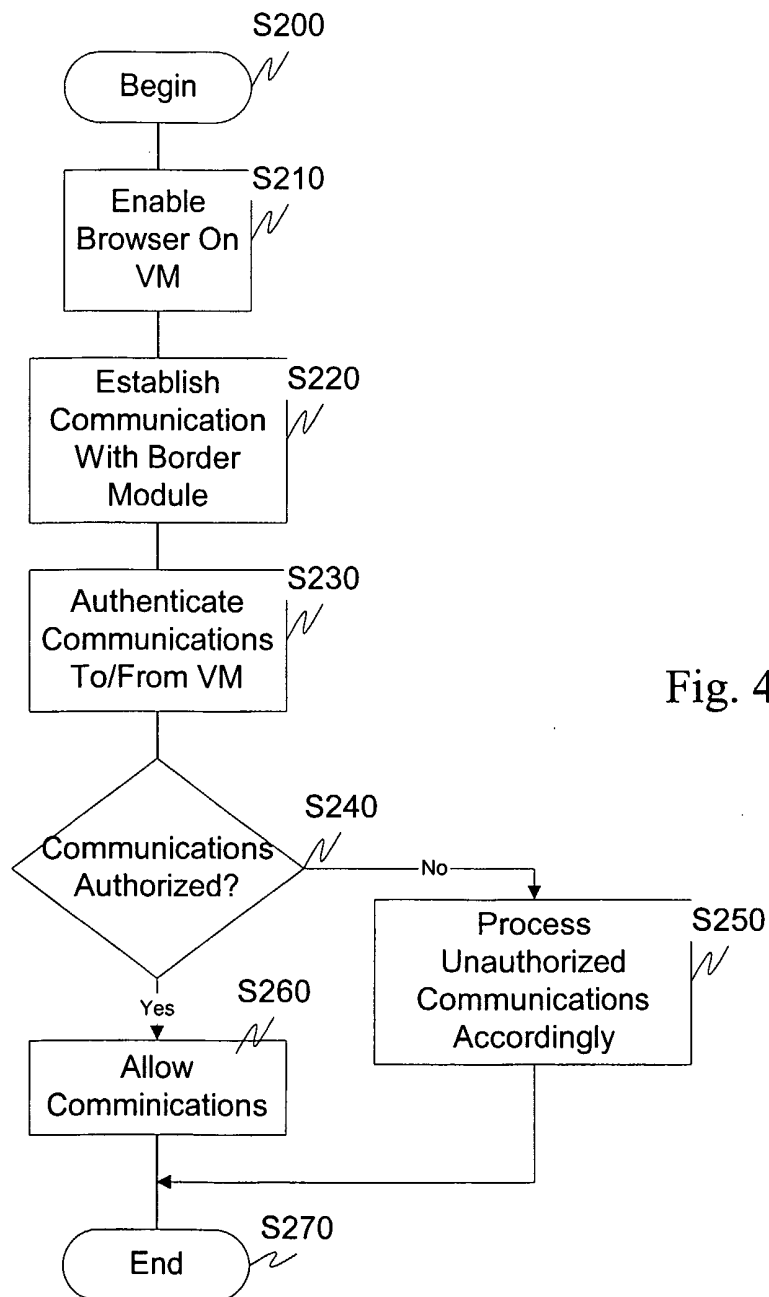


Fig. 4

# INTERNATIONAL SEARCH REPORT

International application No.

PCT/US02/06018

## A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : H04L 9/12

US CL : 713/201

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)

U.S. : 713/201, 150-151, 153, 155 160, 162; 709/217, 218; 707/230, 245, 513

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)  
Please See Continuation Sheet

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category *    | Citation of document, with indication, where appropriate, of the relevant passages                                                                                                                                                                              | Relevant to claim No.                     |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|
| X<br>---<br>Y | US 5,835,726 A (SHWED et al.) 10 November 1998 (10.11.1998), abstract, col. 2, lines 32-36, 41-50, col. 5, lines 36-45, 55-61, col. 6, lines 21-23, see also Fig. 3.                                                                                            | 1-5, 11-13, 17-20<br>-----<br>6-10, 14-16 |
| X<br>---<br>Y | Col. 9, lines 60-67, col. 12, lines 1-17<br>US 6,081,900 A (SUBRAMANIAM et al.) 27 June 2000 (27.06.2000), abstract, ,See Fig. 2, ,Fig.3, col. 5, lines 40-67, col. 5, lines 25-37, col. 6, lines 61-67col. 7, lines 1-20col. 8, lines 1-67col. 10, lines 20-67 | 1-5, 11-20<br>-----<br>6-10, 14-16        |
| Y             | US 6,088,796 A (CIANFROCCA et al.) 11 July 2000 (11.07.2000), col. 4, lines 20-59, col. 5, lines 14-67, col. 6, lines 50-67col. 61, lines 61-67col. 8, lines 40-54col. 12, lines 15-46                                                                          | 1-20                                      |



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                                                                | "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                                              |
| "E" earlier application or patent published on or after the international filing date                                                                                   | "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                                                                     |
| "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) | "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 |
| "O" document referring to an oral disclosure, use, exhibition or other means                                                                                            | "&" document member of the same patent family                                                                                                                                                                                                    |
| "P" document published prior to the international filing date but later than the priority date claimed                                                                  |                                                                                                                                                                                                                                                  |

Date of the actual completion of the international search

16 June 2002 (16.06.2002)

Date of mailing of the international search report

11 JUL 2002

Name and mailing address of the ISA/US

Commissioner of Patents and Trademarks

Box PCT

Washington, D.C. 20231

Facsimile No. (703)305-3230

Authorized officer

Gail O Hayes

Telephone No. (703) 305-4274

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/US02/06018

**Continuation of B. FIELDS SEARCHED Item 3:**

WEST, Dialog, ProQuest, Dogpile. Search Terms: Security and Intranet access, Firwall and unauthorized access, Brwoser and connection, security module and network.