UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/011,022	07/21/2010	5623600	FORT-000013L	3498

26853 7590 01/06/2011

COVINGTON & BURLING, LLP
ATTN: PATENT DOCKETING
1201 PENNSYLVANIA AVENUE, N.W.
WASHINGTON, DC 20004-2401

EXAMINER	
ART UNIT	PAPER NUMBER

DATE MAILED: 01/06/2011

Please find below and/or attached an Office communication concerning this application or proceeding.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

MAILED

JAN 06 2011

CENTRAL REEXAMINATION UNIT

DO NOT USE IN PALM PRINTER

(THIRD PARTY REQUESTER'S CORRESPONDENCE ADDRESS)

Michael A. DeSanctis
Hamilton DeSanctis &Cha LLP
225 Union Blvd. Suite 150
Lakewood, CO 80228

EX PARTE REEXAMINATION COMMUNICATION TRANSMITTAL FORM

REEXAMINATION CONTROL NO. *90/011,022*.

PATENT NO. *5623600*.

ART UNIT *3992*.

Enclosed is a copy of the latest communication from the United States Patent and Trademark Office in the above identified *ex parte* reexamination proceeding (37 CFR 1.550(f)).

Where this copy is supplied after the reply by requester, 37 CFR 1.535, or the time for filing a reply has passed, no submission on behalf of the *ex parte* reexamination requester will be acknowledged or considered (37 CFR 1.550(g)).

Office Action in Ex Parte Reexamination	Control No. 90/011,022	Patent Under Reexamination 5623600	
	Examiner MINH DIEU NGUYEN	Art Unit 3992	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

a☒ Responsive to the communication(s) filed on *21 July 2010*. b☐ This action is made FINAL.
c☒ A statement under 37 CFR 1.530 has not been received from the patent owner.

A shortened statutory period for response to this action is set to expire *2* month(s) from the mailing date of this letter. Failure to respond within the period for response will result in termination of the proceeding and issuance of an *ex parte* reexamination certificate in accordance with this action. 37 CFR 1.550(d). **EXTENSIONS OF TIME ARE GOVERNED BY 37 CFR 1.550(c).** If the period for response specified above is less than thirty (30) days, a response within the statutory minimum of thirty (30) days will be considered timely.

Part I THE FOLLOWING ATTACHMENT(S) ARE PART OF THIS ACTION:

1. ☒ Notice of References Cited by Examiner, PTO-892.
2. ☐ Information Disclosure Statement, PTO/SB/08.
3. ☐ Interview Summary, PTO-474.
4. ☐ _____.

Part II SUMMARY OF ACTION

1a. ☒ Claims *1-22* are subject to reexamination.
1b. ☐ Claims _____ are not subject to reexamination.
2. ☐ Claims _____ have been canceled in the present reexamination proceeding.
3. ☐ Claims _____ are patentable and/or confirmed.
4. ☒ Claims *1-22* are rejected.
5. ☐ Claims _____ are objected to.
6. ☐ The drawings, filed on _____ are acceptable.
7. ☐ The proposed drawing correction, filed on _____ has been (7a)☐ approved (7b)☐ disapproved.
8. ☐ Acknowledgment is made of the priority claim under 35 U.S.C. § 119(a)-(d) or (f).
   a)☐ All b)☐ Some* c)☐ None of the certified copies have
   1☐ been received.
   2☐ not been received.
   3☐ been filed in Application No. _____.
   4☐ been filed in reexamination Control No. _____.
   5☐ been received by the International Bureau in PCT application No. _____.
   * See the attached detailed Office action for a list of the certified copies not received.
9. ☐ Since the proceeding appears to be in condition for issuance of an *ex parte* reexamination certificate except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte* Quayle, 1935 C.D. 11, 453 O.G. 213.
10. ☒ Other: *this is a supplemental non-final rejection including additional references not cited in the previous action. The period for response runs 2 months from the mailing of this action*

cc: Requester (if third party requester)

DETAILED ACTION

I. Procedures Governing Reexamination

Proposed Amendments, Affidavits, or Declarations

In order to ensure full consideration of any amendments, affidavits or declarations, or other documents as evidence of patentability, such documents must be submitted in response to this Office action. Submissions after the next Office action, which is intended to be a final action, will be governed by the requirements of 37 CFR 1.116, after final rejection and 37 CFR 41.33 after appeal, which will be strictly enforced.

Patent owner is notified that any proposed amendment to the specification and/or claims in this reexamination proceeding must comply with 37 CFR 1.530(d)-(j), must be formally presented pursuant to 37 CFR 1.52(a) and (b), and must contain any fees required by 37 CFR 1.20(c).

Extensions of Time

Extensions of time under 37 CFR 1.136(a) will not be permitted in these proceedings because the provisions of 37 CFR 1.136 apply only to "an applicant" and not to parties in a reexamination proceeding. Additionally, 35 U.S.C. 305 requires that reexamination proceedings "will be conducted with special dispatch" (37 CFR 1.550(a)).

Extension of time in *ex parte* reexamination proceedings are provided for in 37 CFR 1.550(c).

Concurrent Litigation

The patent owner is reminded of the continuing responsibility under 37 CFR 1.565(a) to apprise the Office of any litigation activity, or other prior or concurrent proceeding, involving the patent at issue in this reexamination proceeding throughout the course of this reexamination proceeding. The third party requester is also reminded of the ability to similarly apprise the Office of any such activity or proceeding throughout the course of this reexamination proceeding. See MPEP §§ 2207, 2282 and 2286.

II. Summary of the Reexamination Proceeding

The '600 patent was issued on April 22, 1997 from an application filed on September 26, 1995.

a) During the prosecution of the '600 patent, on August 27, 1996, the Examiner initially rejected claim 1 as being obvious over Lerche et al. (U.S. Patent 5,511,163) in view of Hile et al. (U.S. Patent 5,319,776), rejected claims 5, 6-8, 13-15, 17 and 20 as being obvious over Hile et al. (U.S. Patent 5,319,776) in view of Lerche et al. (U.S. Patent 5,511,163) and allowed claims 22, 24-26. Claims 2-4, 9-12, 16, 18-19, 21 and 23 were objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form. The Examiner indicated that "As per claims 2-4, 11, and 12, the prior arts do not teach, singly or in combination, that the server is a proxy server nor do they teach a FTP or SMTP proxy server to handle evaluation and transfer of data files. The prior arts

also fail to teach a daemon for transferring data from the proxy server wherein the daemon is an FTP or SMTP daemon.

As per claims 9 and 10, The prior arts fail to teach, singly or in combination, the step of determining whether the data is of a type and transmitting the data from the server to the destination without performing the steps of scanning, determining, performing and sending, if the data is not of a type that is likely to contain a virus.

As per claim 16, the prior arts fail to teach, singly or in combination, that the server includes a SMTP proxy server and a SMTP daemon.

As per claims 18, 19, and 21, the prior arts fail to teach, singly or in combination, the steps of storing each encoded portion of the mail message (data) in a separate file; decoding the encoded portions of the data (mail message) to produce decoded portions of the mail message; and scanning each of the decoded portions for a virus."

b) On September 24, 1996, in response to the rejection, Applicant amended claim 1 by incorporating objected claim 2, claim 5 (now claim 4) by incorporating objected claim 9, claim 13 (now claim 11) by incorporating objected claim 18, and claim 22 (now claim 18). Claim 16 was rewritten in independent form to create now claim 13.

c) A notice of allowance was then issued on October 22, 1996 without additional comment by the Examiner.

Accordingly, the record suggests that claims 1, 4, 11, 13 and 18 of the '600 patent were issued because the cited prior art failed to teach or suggest the server is a proxy server, FTP or SMTP proxy server to handle evaluation and transfer of data files; a daemon for transferring data from the proxy server wherein the daemon is an FTP or SMTP daemon; determining whether the data contains virus at the server and transmitting the data from the server to the destination without performing the steps of

determining and performing, if the data is not of a type that is likely to contain a virus; the server includes a SMTP proxy server and a SMTP daemon; and storing each encoded portion of the mail message (data) in a separate file, decoding the encoded portions of the data (mail message) to produce decoded portions of the mail message, and scanning each of the decoded portions for a virus.

d) A first request for Reexamination was filed on July 21, 2010 seeking reexamination of claims 1-22 and on September 16, 2010 reexamination was ordered for claims 1-22.

e) A second request for Reexamination was filed on September 21, 2010 seeking reexamination of claims 1-22.

III. Grounds of Rejection

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

Claims 18-20 and 22 are rejected under 35 U.S.C. 102(a) as being anticipated by **Norman Data Defense Systems, Inc., June 1995 (hereafter Norman)** (An Introduction to the Norman Firewall).

a) As to claim 18, Norman discloses an apparatus for detecting viruses in data transfers between a first computer and a second computer (Norman, page 4 – the

firewall includes a fully configured secure computer system and virus detection capability), the apparatus comprising: means for receiving a data transfer request including a destination address (Norman, page 1 - With a proxy server between an internal network and external connections, "IP packets will not pass directly from the input to the output network interfaces", because "the proxy server runs two separate connections with the proxy as the carrier in between". Page 7, the firewall of Norman "uses nothing but proxy services to pass traffic from one network to the other. No packets will be allowed to pass directly." Such a proxy server necessarily receives data transfer requests from internal network nodes. Page 8 - With respect to outgoing transfers, the firewall "log[s] into the workstation on the secure network to transfer the requested file". Page 5 - The firewall "can identify the packets' destination"); means for electronically receiving data at a server (Norman describes a firewall having a proxy server; server that receives incoming data. The proxy server stands "between the [internal] network and any external connections .... IP packets will not pass directly from the input to the output network interfaces in the proxy server environment." (Norman, p. 1.); means for determining whether the data contains a virus at the server (The firewall of Norman "uses a proxy server" (Norman, p. 1) which "automatically checks every incoming file for viruses before letting the file through" (Norman, p. 5); means for performing a preset action on the data using the server if the data contains a virus (The firewall of Norman "scans all incoming files for any of 7100+ viruses, and sets them aside for later examination rather than forwarding them, if they are infected" (Norman, p. 5). "When a virus is located [by the firewall], the file transaction is blocked and logged."(Norman, p. 9.) The firewall "can be made to notify a network management station on the internal net through SNMP traps. If a virus.., is discovered, traps can be sent to one or several machines on the secure network." (Norman, p.20); and means for sending the data to the destination address if the data does not contain a virus (Traffic that is due to be checked for viruses..[is] queued, and the [antivirus] module will then scan and give clearances for each file. When a file is cleared, it is then passed on by the proxy process." (Norman, p. 9.)

b) **As to claim 19**, Norman discloses wherein means for determining includes a means for scanning that scans the data using a signature scanning process (Norman, page 9, states that "[a]s new viruses are discovered and analyzed, their 'signatures' are included in the virus definition file (NVC.DEF)", a files that is updated regularly).

c) **As to claim 20**, Norman discloses wherein the means for performing a preset action comprises: means for transmitting the data unchanged (Transmitting data unchanged, even if it contains a virus, simply represents the ordinary operation of prior art network gateways which performed no antivirus scanning); means for not transmitting the data (The firewall of Norman "scans all incoming files for any of 7100+ viruses, and sets them aside for later examination rather than forwarding them, if they are infected" (Norman, p. 5). "When a virus is located [by the firewall], the file transaction is blocked and logged." (Norman, p. 9); and means for storing the data in a file with a new name and notifying a recipient of the data transfer request of the new file name.

d) **As to claim 22**, Norman discloses further comprising means for determining whether the data is being transferred into a first network by comparing the destination address to valid addresses for the first network (Norman teaches a firewall that "can identify the packets' destination" (Norman, p. 5). Moreover, conventional network security products "'read' the address information in packets and direct each to its intended destination" (Norman, p. 5). For example, a screening router applies rules that "rely on the origin and destination IP-addresses to decide if a packet is 'good' or 'bad' " (Norman, p. 3).

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and

the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-3 are rejected under 35 U.S.C. 103(a) as being obvious over **Cheswick** (The Design of a Secure Internet Gateway) in view of **Cheswick and Bellovin (hereafter CB)** (Firewalls and Internet Security) and further in view of **TIS Firewall** (TIS Firewall Toolkit Overview).

a) **As to claim 1,** Cheswick discloses a system for detecting and selectively removing viruses in data transfers (Cheswick, page 233-234 - The New Gateway, named inet, is used so the internal machines are protected even if an invader breaks into the gateway machine, becomes root and creates and runs a new kernel), the system comprising: a memory for storing data and routines, the memory having inputs and outputs, the memory including a server for scanning data for a virus and specifying data handling actions dependent on an existence of the virus (Cheswick, page 234 – The Inet gateway is a MIPS M/120 running System V with Berkeley enhancements. Various daemons and critical programs have been obtained from other sources, checked and installed, page 235 – Inbound mail is delivered directly to Inet. Inet checks the destination. If it is a trusted machine (i.e. its smtp is trusted), a connection request is sent to r70 (a single internal machine that provides a limited set of services to Inet for reaching internal machines). If not, the mail is relayed through an accessible internal machine); a communications unit for receiving and sending data in response to control signals, the communications unit having an input and an output (Cheswick, page 234 – Cheswick discloses the design of a secure internet gateway for providing incoming login and mail service and outgoing mail, so a communications unit is inherently present in any system for transferring data); a processing unit for receiving signals from the

memory and the communications unit and for sending signals to the memory and communications unit; the processing unit having inputs and outputs; the inputs of the processing unit coupled to the outputs of memory and the output of the communications unit; the outputs of the processing unit coupled to the inputs of memory, the input of the communications unit, the processor controlling and processing data transmitted through the communications unit to detect viruses and selectively transfer data depending on the existence of viruses in the data being transmitted (Cheswick, page 234 – the Inet uses a MIPS M/120 processor on the gateway, the base UNIX operating system, and the inclusion of an Ethernet board to connect to a router. The inclusion of memory and the attachment of memory to a communication process is inherent and obvious in the context of Cheswick).

Cheswick discloses inbound mail is delivered directly to Inet. Inet checks the destination. If it is a trusted machine (i.e. its smtp is trusted), a connection request is sent to r70 (a single internal machine that provides a limited set of services to Inet for reaching internal machines). If not, the mail is relayed through an accessible internal machine (page 235).

Cheswick does not explicitly disclose, however CB discloses the processor controlling and processing data transmitted through the communications unit to detect viruses and selectively transfer data depending on the existence of viruses in the data being transmitted (CB discloses the use of firewalls to significantly increase security on network computers. Chapter 3, pages 51, 70, 75-76 – Packet filtering, circuit gateways, and application gateways are discussed. Commonly, more than one of these is used at the same time to log and control all incoming and outgoing traffic to scan for viruses).

It would have been obvious to one of ordinary skill in the art at the time of the invention to employ the use of having the processor controlling and processing data transmitted through the communications unit to detect viruses and selectively transfer data depending on the existence of viruses in the data being transmitted in the system of Cheswick, as CB discloses, so as to increase security on network computers.

Cheswick and CB disclose a proxy server and a daemon (Cheswick, pages 234-235 – discussing the implementation of a gateway and use of a proxy and various daemons in the context of providing scanning and security services. CB, Chapter 6: Gateway Tools – discussing the use of proxies and daemons as fundamental gateway components to manage network communications and provide network security services, including scanning for viruses and operations to deal with security threats).

Cheswick and CB do not explicitly disclose, however TIS Firewall discloses a proxy server for receiving data to be transferred, the proxy server scanning the data to be transferred for viruses and controlling transmission of the data to be transferred according to preset handing instructions and the presence of viruses, the proxy server having a data input a data output and a control output the data input coupled to receive the data to be transferred (TIS Firewall, pages 3-4 – The toolkit software provides proxy services for common applications like FTP and TELNET, and security for SMTP mail, the toolkit software is configured to address "that which is not expressly permitted is denied);

and a daemon for transferring data from the proxy server in response to control signals from the proxy server, the daemon having a control input, a data input and a data output the control input of the daemon coupled to the control output of the proxy server for receiving control signals, and the data input of the daemon coupled to the data output of the proxy server for receiving the data to be transferred (TIS Firewall, page

10 – The toolkit includes source code for a modified version of the FTP daemon which permits an administrator to provide both FTP service and FTP proxy service on the same system).

It would have been obvious to one of ordinary skill in the art at the time of the invention to employ the use of having a proxy server for receiving data to be transferred, the proxy server scanning the data to be transferred for viruses and controlling transmission of the data to be transferred according to preset handing instructions and the presence of viruses, the proxy server having a data input a data output and a control output the data input coupled to receive the data to be transferred and a daemon for transferring data from the proxy server in response to control signals from the proxy server, the daemon having a control input, a data input and a data output the control input of the daemon coupled to the control output of the proxy server for receiving control signals, and the data input of the daemon coupled to the data output of the proxy server for receiving the data to be transferred in the system of Cheswick and CB, as TIS Firewall discloses, so as to achieve all different levels of security from the basic to the most rigorous security configurations (TIS Firewall, page 1).

b) **As to claim 2,** the combination of Cheswick, CB and TIS Firewall discloses the proxy server is a FTP proxy server that handles evaluation and transfer of data files, and the daemon is an FTP daemon that communicates with a recipient node and transfers data files to the recipient node (TIS Firewall, page 10 - A proxy server for FTP).

c) **As to claim 3,** combination of Cheswick, CB and TIS Firewall discloses

the proxy server is a SMTP proxy server that handles evaluation and transfer of messages, and the daemon is an SMTP daemon that communicates with a recipient node and transfers messages to the recipient node (TIS Firewall, page 8 – SMTP service).

Claims 4 and 7 are rejected under 35 U.S.C. 103(a) as being obvious over **Cheswick and Bellovin (hereafter CB)** (Firewalls and Internet Security) and in view of **Sidewinder** (Special Report: Secure Computing Corporation and Network Security).

a) **As to claim 4,** CB discloses a computer implemented method for detecting viruses in data transfers between a first computer and a second computer, the method comprising the steps of: receiving at a server a data transfer request including a destination address (CB, pages 66-69, 74-75 – CB describes a system that receives data transfer requests with a destination address at a server); electronically receiving data at the server; determining whether the data contains a virus at the server (CB, page 76 – a location with many PC users might wish to scan incoming files for viruses, Chapter 3 "Firewall Gateways" including a discussion of packet filtering, filtering rules and filter placement; also, protocol specific filtering to detect viruses in data transfers); performing a preset action on the data using the server if the data contains a virus (CB, page 76 - Application gateways are often used in conjunction with the other gateway designs, packet filters and circuit-level relays. As we show later [], an application gateway can be used to pass X11 [a type of network traffic] through a firewall with reasonable security. The semantic knowledge inherent in the design of an application gateway can be used in more sophisticated fashions. As described earlier, gopher servers can specify that a file is in the format used by the uuencode program. But that format includes a file name and mode. A clever gateway could examine or even rewrite this line, thus blocking attempts to force the installation of bogus .rhosts files or shells with the setuid bit

turned on. The type of filtering used depends on local needs and customs. A location with many PC users might wish to scan incoming files for viruses).

CB does not explicitly disclose, however Sidewinder discloses certain types of data can be selectively prohibited from passing to and from the external network, by sending the data to the destination address if the data does not contain a virus; determining whether the data is of a type that is likely to contain a virus; and transmitting the data from the server to the destination without performing the steps of determining whether the data contains a virus and performing a preset action if the data is not of a type that is likely to contain a virus (Sidewinder, pages SR-454.9, SR-454-10 – block all incoming and outgoing news which does not fit the statistical properties of English-language plaintext, filter incoming and outgoing news on the basis of content similarity to postings deemed to be in violation of the site's policy. Page SR-454.4 - certain classes of data may be prohibited from passing to and from the external network).

It would have been obvious to one of ordinary skill in the art at the time of the invention to employ the teaching of selectively transfer data based on the existence of viruses within such data by sending the data to the destination address if the data does not contain a virus; determining whether the data is of a type that is likely to contain a virus; and transmitting the data from the server to the destination without performing the steps of determining whether the data contains a virus and performing a preset action if the data is not of a type that is likely to contain a virus in the system of CB, as Sidewinder teaches so as to avoid downstream virus infection.

b) **As to claim 7**, the combination of CB and Sidewinder discloses wherein the step of performing a preset action on the data using the server comprises

performing one step from the group of: transmitting the data unchanged; not transmitting the data; and storing the data in a file with a new name and notifying a recipient of the data transfer request of the new file name (Sidewinder, SR-454.8 – SR-454-12 - messages which fail to pass the filter are passed to the System Administrator for action. Rejected mail may be discarded or kept in a 'trash' folder for later examination. Outgoing data which has been blocked by the filter is forwarded to the System Administrator for disposition. Incoming data which has been blocked by the filter is discarded (i.e. not transmitted).

Claims 5, 8, 11-14 and 16-17 are rejected under 35 U.S.C. 103(a) as being obvious over **Cheswick and Bellovin (hereafter CB)** (Firewalls and Internet Security) in view of **Sidewinder** (Special Report: Secure Computing Corporation and Network Security) and further in view of **MIMEsweeper** (MIMEsweeper administrator guide).

a) **As to claim 5**, the combination of CB and Sidewinder discloses the step of determining includes scanning the data for a virus using the server (CB, page 76 – a location with many PC users might wish to scan incoming files for viruses, Chapter 3 "Firewall Gateways" including a discussion of packet filtering, filtering rules and filter placement; also, protocol specific filtering to detect viruses in data transfers).

The combination of CB and Sidewinder does not explicitly disclose, however MIMEsweeper discloses the steps of storing the data in a temporary file at the server after the step of electronically transmitting (MIMEsweeper, page 13 – "The SMTP server must also store messages, on receipt, in a form and location suitable for MIMEsweeper to read and analyze, and then collect cleared messages for onward delivery).

It would have been obvious to one of ordinary skill in the art at the time of the invention to employ the use of storing the data in a temporary file at the server after the

step of electronically transmitting in the system of CB and Sidewinder, as MIMEsweeper discloses, so as to allow a network administrator or the like to later review and evaluate the transmitted data.

b) **As to claim 8**, the combination of CB, Sidewinder and MIMEsweeper discloses the step of determining whether the data is of a type that is likely to contain a virus is performed by comparing an extension type of a file name for the data to a group or known extension types (MIMEsweeper, page 49 – "The way a file is scanned depends on the type of file...to be scanned and the validator employed").

c) **As to claim 11,** CB discloses a computer implemented method for detecting viruses in a mail message transfers between a first computer and a second computer, the method comprising the steps of: receiving a mail message request including a destination address (CB, pages 66-69, 74-75 – CB describes a system that receives data transfer requests with a destination address at a server); electronically receiving the mail message at a server; determining whether the mail message contains a virus (CB, page 76 – a location with many PC users might wish to scan incoming files for viruses, Chapter 3 "Firewall Gateways" including a discussion of packet filtering, filtering rules and filter placement; also, protocol specific filtering to detect viruses in data transfers); performing a preset action on the mail message if the mail message contains a virus (CB, page 76 - Application gateways are often used in conjunction with the other gateway designs, packet filters and circuit-level relays. As we show later [], an application gateway can be used to pass X11 [a type of network traffic] through a firewall with reasonable security. The semantic knowledge inherent in the design of an application gateway can be used in more sophisticated fashions. As described earlier, gopher servers can specify that a file is in the format used by the uuencode program. But that format includes a file name and mode. A clever gateway could examine or even rewrite this line, thus blocking attempts to force the installation of bogus .rhosts

files or shells with the setuid bit turned on. The type of filtering used depends on local needs and customs. A location with many PC users might wish to scan incoming files for viruses).

CB does not explicitly disclose, however Sidewinder discloses sending the mail message to the destination address if the mail message does not contain a virus; (Sidewinder, pages SR-454.9, SR-454-10 – block all incoming and outgoing news which does not fit the statistical properties of English-language plaintext, filter incoming and outgoing news on the basis of content similarity to postings deemed to be in violation of the site's policy. Page SR-454.4 - certain classes of data may be prohibited from passing to and from the external network).

It would have been obvious to one of ordinary skill in the art at the time of the invention to employ sending the mail message to the destination address if the mail message does not contain a virus in the system of CB, as Sidewinder teaches so as to avoid downstream virus infection.

The combination of CB and Sidewinder does not disclose, however MIMEsweeper discloses the determination of whether the mail message contains a virus comprising determining whether the mail message includes any encoded portions, storing each encoded portion of the mail message in a separate temporary file, decoding the encoded portions of the mail message to produced decoded portions of the mail message, scanning each of the decoded portions for a virus, and testing whether the scanning step found any viruses (MIMEsweeper discloses a total E-mail content management tool. It breaks the message into its constituent elements and then subjects each of those components to different checks depending on the content. Page 9 - "MIMEsweeper provides a framework for total Email content management. Once MIMEsweeper is configured into Email routing it can analyze the content of each message. MIMEsweeper breaks the messages into its constituent elements and then subjects each of those components to different checks depending on content". The

MIMEsweeper extracts the elements from the mail messages and then presents all the extracted elements to external programs for analysis. MIMEsweeper is recursive in its analysis; so it will find a ZIP file within a ZIP file and a uuencoded component of that file. In other words the analysis continues until MIMEsweeper cannot break the message down further". "The rationale behind this is that Email borne threats might not be recognized by checks if they are compressed or encoded".

It would have been obvious to one of ordinary skill in the art at the time of the invention to employ the use of determination of whether the mail message contains a virus comprising determining whether the mail message includes any encoded portions, storing each encoded portion of the mail message in a separate temporary file, decoding the encoded portions of the mail message to produced decoded portions of the mail message, scanning each of the decoded portions for a virus, and testing whether the scanning step found any viruses in the system of CB and Sidewinder, as MIMEsweeper discloses so as to selectively transfer data based on the existence of viruses in order to avoid downstream virus infection.

d) **As to claim 12**, the combination of CB, Sidewinder and MIMEsweeper discloses wherein the step of determining whether the mail message includes any encoded portions searches for uuencoded portions (MIMEsweeper, page 9 - MIMEsweeper is recursive in its analysis; so it will find a ZIP file within a ZIP file and a uuencoded component of that file).

e) **As to claim 13,** CB discloses a computer implemented method for detecting viruses in a mail message transfers between a first computer and a second computer, the method comprising the steps of: receiving a mail message request including a destination address (CB, pages 66-69, 74-75 – CB describes a system that receives data transfer requests with a destination address at a server); electronically receiving the mail

message at a server; determining whether the mail message contains a virus (CB, page 76 – a location with many PC users might wish to scan incoming files for viruses, Chapter 3 "Firewall Gateways" including a discussion of packet filtering, filtering rules and filter placement; also, protocol specific filtering to detect viruses in data transfers); performing a preset action on the mail message if the mail message contains a virus (CB, page 76 - Application gateways are often used in conjunction with the other gateway designs, packet filters and circuit-level relays. As we show later [], an application gateway can be used to pass X11 [a type of network traffic] through a firewall with reasonable security. The semantic knowledge inherent in the design of an application gateway can be used in more sophisticated fashions. As described earlier, gopher servers can specify that a file is in the format used by the uuencode program. But that format includes a file name and mode. A clever gateway could examine or even rewrite this line, thus blocking attempts to force the installation of bogus .rhosts files or shells with the setuid bit turned on. The type of filtering used depends on local needs and customs. A location with many PC users might wish to scan incoming files for viruses).

CB does not explicitly disclose, however Sidewinder discloses sending the mail message to the destination address if the mail message does not contain a virus; (Sidewinder, pages SR-454.9, SR-454-10 – block all incoming and outgoing news which does not fit the statistical properties of English-language plaintext, filter incoming and outgoing news on the basis of content similarity to postings deemed to be in violation of the site's policy. Page SR-454.4 - certain classes of data may be prohibited from passing to and from the external network).

It would have been obvious to one of ordinary skill in the art at the time of the invention to employ sending the mail message to the destination address if the mail message does not contain a virus in the system of CB, as Sidewinder teaches so as to avoid downstream virus infection.

The combination of CB and Sidewinder does not disclose, however MIMEsweeper discloses scanning the mail message for encoded portions, wherein the

step of sending the mail message to the destination address is performed if the mail message does not contain any encoded portions; the server includes a SMTP proxy server and a SMTP daemon; and the step of sending the mail message comprises transferring the mail message from the SMTP proxy server to the SMTP daemon and transferring the mail message from the SMTP daemon to a node having an address matching the destination address (MIMEsweeper discloses a total E-mail content management tool. It breaks the message into its constituent elements and then subjects each of those components to different checks depending on the content. Page 9 - "MIMEsweeper provides a framework for total Email content management. Once MIMEsweeper is configured into Email routing it can analyze the content of each message. MIMEsweeper breaks the messages into its constituent elements and then subjects each of those components to different checks depending on content". The MIMEsweeper extracts the elements from the mail messages and then presents all the extracted elements to external programs for analysis. MIMEsweeper is recursive in its analysis; so it will find a ZIP file within a ZIP file and a uuencoded component of that file. In other words the analysis continues until MIMEsweeper cannot break the message down further". "The rationale behind this is that Email borne threats might not be recognized by checks if they are compressed or encoded". Page 13, "The client-server architecture of SMTP mail means that a fully functional SMTP server is required to handle the receipt of Email items from the Internet, and their delivery to local or remote users after MIMEsweeper checking. The SMTP server must also store messages, on receipt, in a form and location suitable for MIMEsweeper to read and analyze, and then collect cleared messages for onward delivery" MIMEsweeper teaches scanning the incoming email attachments for the presence of computer viruses. The architecture involved incorporates a message store for storing the messages temporarily. The MIMEsweeper operates while transferring the data between the message stores. Page 10 ("MIMEsweeper as mail transfer agent"). The MIMEsweeper firstly reads a waiting message from the database, analyzes its contents, and then depending on the analysis, it submits the message for onward transmission or diverts it according to a quarantine policy. MIMEsweeper 'quarantines' any mail message found to contain a virus or unidentifiable attachment based

on the assumption that viruses can be in any part of an attachment. Page 7 ("MIMEsweeper takes a holistic approach in that it assumes viruses can be in any part of an attachment. Any mail message found to contain a virus or unidentifiable attachment is 'quarantined'. The configurable nature of MIMEsweeper also allows the quarantining of other user-specified file types").

It would have been obvious to one of ordinary skill in the art at the time of the invention to employ the use of scanning the mail message for encoded portions, wherein the step of sending the mail message to the destination address is performed if the mail message does not contain any encoded portions; the server includes a SMTP proxy server and a SMTP daemon; and the step of sending the mail message comprises transferring the mail message from the SMTP proxy server to the SMTP daemon and transferring the mail message from the SMTP daemon to a node having an address matching the destination address in the system of CB and Sidewinder, as MIMEsweeper discloses so as to selectively transfer data based on the existence of viruses in order to avoid downstream virus infection.

f) **As to claim 14,** the combination of CB, Sidewinder and MIMEsweeper discloses wherein the step of determining whether the mail message contains a virus, further comprises the steps of: storing the message in a temporary file; scanning the temporary file for viruses; and testing whether the scanning step found a virus.

g) **As to claim 16**, the combination of CB, Sidewinder and MIMEsweper discloses wherein the step of performing a preset action on the mail message using the server comprises performing one step from the group of: transmitting the mail message unchanged; not transferring the mail message; storing the mail message as a file with a new name and notifying a recipient of the mail message request of the new file name;

and creating a modified mail message by writing the output of the determining step into the modified mail message and transferring the mail message to the destination address (Sidewinder, SR-454.8 – SR-454-12 - messages which fail to pass the filter are passed to the System Administrator for action. Rejected mail may be discarded or kept in a 'trash' folder for later examination. Outgoing data which has been blocked by the filter is forwarded to the System Administrator for disposition. Incoming data which has been blocked by the filter is discarded (i.e. not transmitted).

h) **As to claim 17**, the combination of CB, Sidewinder and MIMEsweper discloses wherein the step of performing a preset action on the mail message comprises performing one step from the group of: transferring the mail message unchanged; transferring the mail message with the encoded portions having a virus deleted; and renaming the encode portions of the mail message containing a virus, and storing the renamed portions as files in a specified directory on the server and notifying a recipient of the renamed files and directory; and writing the output of the determining step into the mail message in place of respective encoded portions that contain a virus to create a modified mail message and sending the modified mail message (Sidewinder, SR-454.8 – SR-454-12 - messages which fail to pass the filter are passed to the System Administrator for action. Rejected mail may be discarded or kept in a 'trash' folder for later examination. Outgoing data which has been blocked by the filter is forwarded to the System Administrator for disposition. Incoming data which has been blocked by the filter is discarded (i.e. not transmitted).

Claims 6 and 15 are rejected under 35 U.S.C. 103(a) as being obvious over **Cheswick and Bellovin (hereafter CB)** (Firewalls and Internet Security) in view of **Sidewinder** (Special Report: Secure Computing Corporation and Network Security) in

view of **MIMEsweeper** (MIMEsweeper administrator guide) and further in view of **TIS Firewall** (TIS Firewall Toolkit Overview).

The combination of CB, Sidewinder and MIMEsweeper does not disclose, however TIS Firewall discloses the step of scanning is performed using a signature scanning process (TIS Firewall, page 41 – since many attacks "have a distinctive signature, smap or the firewall's mailer can be configured to attempt to identify these letterbombs").

It would have been obvious to one of ordinary skill in the art at the time of the invention to employ the use of having the step of scanning is performed using a signature scanning process in the system of CB, Sidewinder and MIMEsweeper, as TIS Firewall discloses so as to identify the existence of viruses.

Claims 9-10 are rejected under 35 U.S.C. 103(a) as being obvious over **Cheswick and Bellovin (hereafter CB)** (Firewalls and Internet Security) in view of **Sidewinder** (Special Report: Secure Computing Corporation and Network Security) and further in view of **TIS Firewall** (TIS Firewall Toolkit Overview).

The combination of CB and Sidewinder does not disclose, however TIS Firewall discloses determining whether the data is being transferred into a first network by comparing the destination address to valid addresses for the first network; wherein the server is a FTP proxy server (TIS Firewall, page 41 - The FTP application gateway is a single process that mediates FTP connections between two networks); wherein the step of electronically receiving data comprises the steps of transferring the data from a client node to the FTP proxy server, if the data is not being transferred into the first network (TIS Firewall, page 41 - The FTP application gateway is a single process that mediates FTP connections between two networks.

Routers can control traffic at an IP level, by selectively permitting or denying traffic based on source/destination address or port. Hosts can control traffic at an application level, forcing traffic to move out of the protocol layer for more detailed examination); and wherein the step of electronically receiving data comprises the steps of transferring the data from a server task to an FTP daemon, and then from the FTP daemon to the FTP proxy server if the data is being transferred into the first network (TIS Firewall, page 41 - The FTP application gateway is a single process that mediates FTP connections between two networks. Routers can control traffic at an IP level, by selectively permitting or denying traffic based on source/destination address or port. Hosts can control traffic at an application level, forcing traffic to move out of the protocol layer for more detailed examination. As an example, the FTP proxy can block FTP export of files while permitting import of files, representing a granularity of control that router-based firewalls cannot presently achieve).

It would have been obvious to one of ordinary skill in the art at the time of the invention to employ the use of determining whether the data is being transferred into a first network by comparing the destination address to valid addresses for the first network; wherein the server is a FTP proxy server; wherein the step of electronically receiving data comprises the steps of transferring the data from a client node to the FTP proxy server, if the data is not being transferred into the first network and wherein the step of electronically receiving data comprises the steps of transferring the data from a server task to an FTP daemon, and then from the FTP daemon to the FTP proxy server if the data is being transferred into the first network in the system of CB and Sidewinder, as TIS Firewall teaches so as to facilitate secure outbound and inbound file transfers using a common file transfer mechanism.

Claims 1-3 and 13 are rejected under 35 U.S.C. 103(a) as being obvious over **Norman Data Defense Systems, Inc., June 1995 (hereafter Norman)** (An Introduction to the Norman Firewall) in view of **TIS Firewall** (TIS Firewall Toolkit Overview).

a) **As to claim 1,** Norman discloses a system for detecting and selectively removing viruses in data transfers (Norman, page 4 – Norman teaches a firewall that "include[es] a fully configured secure computer system and virus detection capability" and "provides a single, highly secured route for data to travel between your network and the internet), the system comprising: a memory for storing data and routines, the memory having inputs and outputs, the memory including a server for scanning data for a virus and specifying data handling actions dependent on an existence of the virus (The firewall of Norman is a computing device with memory (RAM); it uses a proxy server, necessarily loaded in memory while running (Norman, pp. 1, 7, 11). "The default configuration is a 100 MHz Intel 486 with 16 MB of RAM and a 1 GB SCSI disk subsystem that is running the SecureWare OS with the firewall software. The other CPU, the front end server, is by default a 66 MHz Intel 486 with 8 MB of RAM and a 500MB hard drive." (Norman, p. 7.) The firewall "has been equipped with an antivirus scanner" that "utilizes the well-known NORMAN Anti-Virus scanner engine, which scans for more then 7100 known viruses ....When a virus is located, the file transaction is blocked and logged." (Norman, p. 9.) The firewall "automatically checks every incoming file for viruses before letting the file through"; it "scans all incoming files for any of 7100+ viruses, and sets them aside for later examination rather than forwarding them, if they are infected" (Norman, p. 5); a communications unit for receiving and sending data in response to control signals, the communications unit having an input and an output ("Nearly all [internet security products] perform addressing, routing and filtering of data packets. They 'read' the address information in packets output; and direct each to the intended destination." (Norman, p. 5.) For example, a screening router

"filter[s] packets using a pre-defined set of rules .... The router then determines whether or not a packet is allowed to pass .... [R]ules can be applied to the source and destination ports.... One can also specify separate sets of rules on incoming and outgoing connections." (Norman, p. 3.) " [A] packet filtering router controls packets at a low level .... Each packet resides on the system for a short moment while the header information is analyzed against the pre-determined rules." (Norman, p. 4.) The firewall of Norman "[a]ttaches LANs to internet via dial-up or dedicated 56 KB or T1 facilities" (Norman, p. 11). "Not merely a packet filter or a router, [it] combines multiple secure computing and communications devices in a single package .... This fully configurable system is tunable to provide the functionality your work demands and the security your organization needs." (Norman, p. 4.); a processing unit for receiving signals from the memory and the communications unit and for sending signals to the memory and communications unit ; the processing unit having inputs and outputs; the inputs of the processing unit coupled to the outputs of memory and the output of the communications unit; the outputs of the processing unit coupled to the inputs of memory, the input of the communications unit, the processor controlling and processing data transmitted through the communications unit to detect viruses and selectively transfer data depending on the existence of viruses in the data being transmitted ("Up to four separate CPUs can be accommodated on the bus. In the basic configuration, two CPUs are supplied. One processor runs the SecureWare operating system. This platform also runs the proxy processes and the anti-virus module. The other processor acts as the un-secure front-end server, and can be configured by the customer." (Norman, p. 6.) "The default configuration is a 100 MHz Intel 486 with 16 MB of RAM and a 1 GB SCSI disk subsystem that is running the SecureWare OS with the firewall software. The other CPU, the front end server, is by default a 66 MHz Intel 486 with 8 MB of RAM and a 500 MB hard drive." (Norman, p. 7.) "A separate Anti-Virus/Hotword process runs on the SecureWare platform. Traffic that is due to be checked for viruses and hotwords are queued, and the module will then scan and give clearances for each file. When a file is

cleared, it is then passed on by the proxy process. (Norman, p. 9.); a proxy server for receiving data to be transferred, the proxy server scanning the data to be transferred for viruses and controlling transmission of the data to be transferred according to preset handing instructions and the presence of viruses, the proxy server having a data input a data output and a control output the data input coupled to receive the data to be transferred ("A superior way of securing an IP network is to apply a so-called proxy server between the network and any external connections ....[T]he proxy machine runs two separate connections with the proxy as a carrier in between. This means that IP packets will not pass directly from the input to the output network interfaces in the proxy server environment:" (Norman, p. 1.) "A more secure approach than packet filtering and routing is the use of so-called proxy processes to convey the traffic between the inside and the outside net. All traffic will then be divided into two separate sessions. One session is established between the internal user and the firewall, and one session is established between the firewall and the external host." (Norman, p. 4.) The firewall of Norman "uses a proxy server" (Norman, p. 1); it "uses nothing but proxy services to pass traffic from one network to the other" (Norman, p. 7)).

Norman does not explicitly disclose, however TIS Firewall discloses a daemon for transferring data from the proxy server in response to control signals from the proxy server, the daemon having a control input, a data input and a data output the control input of the daemon coupled to the control output of the proxy server for receiving control signals, and the data input of the daemon coupled to the data output of the proxy server for receiving the data to be transferred (TIS Firewall teaches a firewall design in which a sendmail proxy communicates with the SMTP daemon (sendmail server), in order to prevent direct network access to sendmail. "This sendmail-proxy, called smap,.., simply accepts all incoming messages and writes them to disk in a spool area .... A second process is responsible for scanning the spool area and delivering the mail messages to the real sendmail for delivery .... Smap preserves sendmail's

functionality, while preventing an arbitrary user on the network from communicating directly with it." (TIS Firewall, p. 41). TIS Firewall also discloses more generally that "[a] proxy forwarder for a network protocol is an application that runs on a firewall host and connects specific service requests across the firewall, acting as a gateway .... Proxies can give the illusion to the software on both sides of a direct point-to-point connection." (TIS Firewall, page 37). The diagram of a telnet application proxy on page 38 of TIS Firewall shows that an application proxy is distinct from, and communicates with, an application daemon (telnetd server).

It would have been obvious to one of ordinary skill in the art at the time of the invention to employ the use of having a daemon for transferring data from the proxy server in response to control signals from the proxy server, the daemon having a control input, a data input and a data output the control input of the daemon coupled to the control output of the proxy server for receiving control signals, and the data input of the daemon coupled to the data output of the proxy server for receiving the data to be transferred in the system of Norman, as TIS Firewall discloses, so as to allow secure network protocol services as well as reuse of existing facilities for data transfer.

b) As to claim 2, the combination of Norman and TIS Firewall discloses the proxy server is a FTP proxy server that handles evaluation and transfer of data files, and the daemon is an FTP daemon that communicates with a recipient node and transfers data files to the recipient node Norman describes a proxy server that includes server is a FTP proxy server that handles proxy services for FTP (Norman, pp. 8, 11). The evaluation and transfer of data files, and the figure in Norman, page 8, illustrates an FTP daemon is an FTP daemon that communicates transaction handled by the firewall. The two-way arrow between the workstation in the protected LAN and the proxy, and the two-way arrow between the proxy and the remote host, demonstrate FTP communication with, and transfers of files to, a recipient node. TIS Firewall teaches a

host-based application-level firewall design in which an FTP proxy controls the transfer of data files between an FTP daemon and a recipient node. A "bastion host provides application-level control" (TIS Firewall, p. 39). "The FTP application gateway is a single process that mediates FTP connections between two networks." (TIS Firewall, p. 41.) "To control FTP access, the application gateway reads a configuration file, containing a list of FTP commands that should be logged, and a description of what systems are allowed to engage in FTP traffic." (TIS Firewall, pp. 41-42). Regarding proxies generally, TIS Firewall states that "[a] proxy for a network protocol is an application that runs on a firewall host and connects specific service requests across the firewall, acting as a gateway .... Proxies can give the illusion to the software on both sides of a direct point-to-point connection. Since many proxies interpret the protocol that they manage, additional access control and audit may be performed as desired. As an example, the FTP proxy can block FTP export of files while permitting import of files, representing a granularity of control that router-based firewalls cannot presently achieve." (TIS Firewall, p. 37.). Although the diagram of an application proxy on page 38 of TIS Firewall is specific to telnet rather than FTP, it shows that an application proxy is distinct from, and communicates with, an application daemon (telnetd server). TIS Firewall discloses the use of an FTP daemon ("common programs such as the FTP server, ftpd") in discussing the advantages of a proxy-based firewall design (TIS Firewall, p. 38; the WUArchive ftpd is referenced on p. 44 as an "FTP server daemon").

c) **As to claim 3,** the combination of Norman and TIS Firewall discloses the proxy server is a SMTP proxy server that handles evaluation and transfer of messages, and the daemon is an SMTP daemon that communicates with a recipient node and transfers messages to the recipient node (Norman describes a proxy server that includes proxy services for SMTP (Norman, pp. 8, 11). TIS Firewall teaches a firewall design in which a sendmail proxy communicates with the SMTP daemon (sendmail server), in order to prevent direct network access to sendmail. "This sendmail-proxy, called smap,.., simply accepts all incoming messages and writes them to disk in a spool area .... A second process is responsible for scanning the spool area and delivering the mail messages to the real sendmail for delivery .... Smap preserves sendmail's functionality, while

preventing an arbitrary user on the network from communicating directly with it." (TIS Firewall, p. 41.) TIS Firewall also discloses more generally that "[a] proxy forwarder for a network protocol is an application that runs on a firewall host and connects specific service requests across the firewall, acting as a gateway .... Proxies can give the illusion to the software on both sides of a direct point-to-point connection. Since many proxies interpret the protocol that they manage, additional access control and audit may be performed as desired." (TIS Firewall, p. 37.) Although the diagram of an application proxy on page 38 of TIS Firewall is specific to telnet rather than FTP, it shows that an application proxy is distinct from, and communicates with, an application daemon (telnetd server)).

d) **As to claim 13,** Norman discloses a computer implemented method for detecting viruses in a mail message transfers between a first computer and a second computer (Norman teaches a firewall which, unlike a mere packet filter or a router, "combines multiple secure computing and communications devices in a single package, including a fully configured secure computer system and virus detection capability" (Norman, p. 4). The firewall "automatically checks every incoming file for viruses before letting the file through" (Norman, p. 5). Incoming files include mail messages being transferred: the firewall "has proxy services for ... SMTP (e-mail)" (Norman, p. 8), the firewall runs mail forwarding software (Norman, p. 6), and the antivirus module acts on contents of electronic mail (Norman, p. 9), the method comprising the steps of: receiving a mail message request including a destination address; electronically receiving the mail message at a server; scanning the mail message for encoded portion; determining whether the mail message contains a virus (With a proxy server between an internal network and external connections, "IP packets will not pass directly from the input to the output network interfaces", because "the proxy server runs two separate connections with the proxy as the carrier in between" (Norman, p. 1). The firewall of Norman "uses nothing but proxy services to pass traffic from one network to the other. No packets will be allowed to pass directly." (Norman, p. 7.) Such a proxy server necessarily receives data transfer requests from internal network nodes. With respect to outgoing transfers, the firewall "log[s] into

the workstation on the secure network to transfer the requested file" (Norman, p. 8). The firewall "can identify the packets' destination" (Norman, p. 5). Internet security products in general "'read' the address information in packets and direct each to its intended destination" (Norman, p. 5). Such devices employ packet routing and filtering, including on outgoing traffic: for example, screening router rules "rely on the origin and destination IP addresses to decide if a packet is 'good' or 'bad' "; "rules can be applied to the source and destination ports", and one can "specify separate sets of rules on incoming and outgoing connections" (Norman, p. 3). Norman describes a firewall having a proxy server that receives incoming data. The proxy server stands "between the [internal] network and any external connections .... IP packets will not pass directly from the input to the output network interfaces in the proxy server environment." (Norman, p. 1.) Norman indicates that uuencoded files will be decoded before being scanned for viruses: "Files that are compressed using one of several known methods, will be uncompressed before scan. Methods currently supported include...UUencode." (Norman, p. 9.));

performing a preset action on the mail message if the mail message contains a virus

The firewall of Norman "scans all incoming files for any of 7100+ viruses, and sets them aside for later examination rather than forwarding them, if they are infected" (Norman, p. 5). "When a virus is located [by the firewall], the file transaction is blocked and logged." (Norman, p. 9.) The firewall "can be made to notify a network management station on the internal net through SNMP traps. If a virus.., is discovered, traps can be sent to one or several machines on the secure network." (Norman, p. 10)); sending the mail message to the destination address if the mail message does not contain a virus;

"A network administrator may also want to control the contents of electronic mail .... Traffic that is due to be checked for viruses and hotwords are queued, and the Antivirus/Hotword module will then scan and give clearances for each file. When a file is cleared, it is then passed on by the proxy process." (Norman, p. 9.).

Norman describes a proxy server that includes proxy services for SMTP (Norman, pages 8, 11), however Norman does not explicitly disclose, and TIS Firewall

discloses wherein the step of sending the mail message to the destination address is performed if the mail message does not contain any encoded portions; the server includes a SMTP proxy server and a SMTP daemon; and the step of sending the mail message comprises transferring the mail message from the SMTP proxy server to the SMTP daemon and transferring the mail message from the SMTP daemon to a node having an address matching the destination address (TIS Firewall teaches a firewall design in which a sendmail proxy communicates with the SMTP daemon (sendmail server), in order to prevent direct network access to sendmail. "This sendmail-proxy, called smap,.., simply accepts all incoming messages and writes them to disk in a spool area .... A second process is responsible for scanning the spool area and delivering the mail messages to the real sendmail for delivery [to the destination address]. Smap preserves sendmail's functionality, wile preventing an arbitrary user on the network from communicating directly with it." (TIS Firewall, page 41). TIS Firewall discloses more generally that "[a] proxy forwarder for a network protocol is an application that runs on a firewall host and connects specific service requests across the firewall, acting as a gateway .... Proxies can give the illusion to the software on both sides of a direct point-to-point connection. Since many proxies interpret the protocol that they manage, additional access control and audit may be performed as desired." (TIS Firewall, p. 37.) Although the diagram of an application proxy on page 38 of TIS Firewall is specific to telnet rather than FTP, it shows that an application proxy is distinct from, and communicates with, an application daemon (telnetd server).

It would have been obvious to one of ordinary skill in the art at the time of the invention to employ the use of wherein the step of sending the mail message to the destination address is performed if the mail message does not contain any encoded portions; the server includes a SMTP proxy server and a SMTP daemon; and the step of sending the mail message comprises transferring the mail message from the SMTP

proxy server to the SMTP daemon and transferring the mail message from the SMTP daemon to a node having an address matching the destination address in the system of Norman, as TIS Firewall discloses so as to secure mail transfer as well as reuse of existing sendmail facilities.

Claims 4, 7-8 and 21 are rejected under 35 U.S.C. 103(a) as being obvious over **Norman Data Defense Systems, Inc., June 1995 (hereafter Norman)** (An Introduction to the Norman Firewall) in view of **David J. Stang, (hereafter Stang)** (ICSA's Computer Virus Handbook).

a) **As to claim 4,** Norman discloses a computer implemented method for detecting viruses in data transfers between a first computer and a second computer (Norman teaches a firewall, "based upon off-the shelf PC-compatible hardware" (Norman, p. 6), that "provides a single, highly secured route for data to travel between your network and the internet" (Norman, p. 4). The firewall "include[es] a fully configured secure computer system and virus detection capability" (Norman, p. 4), the method comprising the steps of: receiving at a server a data transfer request including a destination address With a proxy server between an internal network and external connections, "IP packets will not pass directly from the input to the output network interfaces", because "the proxy server runs two separate connections with the proxy as the carrier in between" (Norman, p. 1). The firewall of Norman "uses nothing but proxy services to pass traffic from one network to the other. No packets will be allowed to pass directly." (Norman, p. 7.) Such a proxy server necessarily receives data transfer requests from internal network nodes. With respect to outgoing transfers, the firewall "log[s] into the workstation on the secure network to transfer the requested file" (Norman, p. 8). The firewall "can identify the packets' destination" (Norman, p. 5). Internet security products in general "'read' the address information in packets and direct each to its intended

destination" (Norman, p. 5). Such devices employ packet routing and filtering, including on outgoing traffic: for example, screening router rules "rely on the origin and destination IP addresses to decide if a packet is 'good' or 'bad' "; "rules can be applied to the source and destination ports", and one can "specify separate sets of rules on incoming and outgoing connections" (Norman, p. 3); electronically receiving data at the server (Norman describes a firewall having a proxy server that receives incoming data. The proxy server stands "between the [internal] network and any external connections .... IP packets will not pass directly from the input to the output network interfaces in the proxy server environment." (Norman, p. 1.); determining whether the data contains a virus at the server (The firewall of Norman "uses a proxy server" (Norman, p. 1) which "automatically checks every incoming file for viruses before letting the file through" (Norman, p. 5); performing a preset action on the data using the server if the data contains a virus (The firewall of Norman "scans all incoming server if the data contains a virus; files for any of 7100+ viruses, and sets them aside for later examination rather than forwarding them, if they are infected" (Norman, p. 5). "When a virus is located [by the firewall], the file transaction is blocked and logged." (Norman, p. 9.) The firewall "can be made to notify a network management station on the internal net through SNMP traps. If a virus.., is discovered, traps can be sent to one or several machines on the secure network." (Norman, p. 20.)); sending the data to the destination address if the data does not contain a virus ("Traffic that is due to be checked for viruses...[is] queued, and the [antivirus] module will then scan and give clearances for each file. When a file is cleared, it is then passed on by the proxy process." (Norman, p. 9.).

Norman does not disclose, however Stang discloses determining whether the data is of a type that is likely to contain a virus; and transmitting the data from the server to the destination without performing the steps of determining whether the data contains a virus and performing a preset action if the data is not of a type that is likely to contain

a virus (Stang explains that virus-infected files are likely to be MS-DOS executable files with particular file extensions. "Once in the machine, the virus does nothing until the program it is attached to is 'run'. At that moment, what it does depends entirely on the species in question. The simpler viruses set out to make copies of themselves in other 'executable' files they can find, increasing the size of those files slightly. Such executable files include any file ending with .EXE, .COM, .OVL, .SYS, or .BIN." (Stang, p. 54.) "Of the hundreds of files on your hard disk, viruses only infect those files that end with the extensions COM and EXE (and sometimes BIN, SYS, OVL, OVR, etc)." (Stang, p. 114.) Transmitting data from the server to the destination, without performing virus detection, simply represents the operation of prior art network gateways. Therefore, it would have been obvious at the time the invention was made to a person having ordinary skill in the art to have a proxy server follow prior art practices by transmitting data without performing virus detection if, using the technique suggested by Stang, the data was determined not to be likely to contain a virus).

It would have been obvious to one of ordinary skill in the art at the time of the invention to employ determining whether the data is of a type that is likely to contain a virus; and transmitting the data from the server to the destination without performing the steps of determining whether the data contains a virus and performing a preset action if the data is not of a type that is likely to contain a virus in the system of Norman, as Stang teaches, so as to reduce the amount of data to be scanned for viruses and minimize delays in transmission of network traffic.

b) As to claim 7, the combination of Norman and Stang discloses wherein the step of performing a preset action on the data using the server comprises performing one step from the group of: transmitting the data unchanged (Transmitting data unchanged, even if it contains a virus, simply represents the ordinary operation of prior art network gateways which performed no antivirus scanning); not transmitting the data (The firewall of Norman

"scans all incoming files for any of 7100+ viruses, and sets them aside for later examination rather than forwarding them, if they are infected" (Norman, p. 5). "When a virus is located [by the firewall], the file transaction is blocked and logged." (Norman, p. 9.); storing the data in a file with a new name and notifying a recipient of the data transfer request of the new file name ("The [firewall] system can even be configured to record the contents of packets and to store sus-pect packets for later review by a security officer." (Norman, p. 5.) At the time the invention was made a person having ordinary skill in the art would have readily appreciated that stored packet contents could be given a unique file name, and that the firewall system could notify the recipient of the file name to allow the recipient to request access to the file).

c) **As to claim 8,** the combination of Norman and Stang discloses wherein the step of determining whether the data is of a type that is likely to contain a virus is performed by comparing an extension type of a file name for the data to a group or known extension types (Stang explains that virus-infected files are likely to be MS-DOS executable files with particular file extensions. "Once in the machine, the virus comparing an extension type of a file name for does nothing until the program it is attached to is 'run'. At that moment, what it does depends entirely on the species in question. The simpler viruses set out to make copies of themselves in other 'executable' files they can find, increasing the size of those files slightly. Such executable files include any file ending with .EXE, .COM, .OVL, .SYS, or .BIN." (Stang, p. 54.) "Of the hundreds of files on your hard disk, viruses only infect those files that end with the extensions COM and EXE (and sometimes BIN, SYS, OVL, OVR, etc)." (Stang, p. 114.)).

d) **As to claim 21,** Norman does not disclose, however Stang disclose a second means for determining whether the data is of a type that is likely to contain a virus (Stang explains that virus-infected files are likely to be MS-DOS executable files with particular file extensions. "Once in the machine, the virus does nothing until the program it is attached to is 'run'. At that moment, what it does depends entirely on the species in question. The simpler viruses set out to make

copies of themselves in other 'executable' files they can find, increasing the size of those files slightly. Such executable files include any file ending with .EXE, .COM, .OVL, .SYS, or .BIN." (Stang, p. 54.) "Of the hundreds of files on your hard disk, viruses only infect those files that end with the extensions COM and EXE (and sometimes BIN, SYS, OVL, OVR, etc)." (Stang, p. 114) ; and means for transmitting the data from the server to the destination without performing the steps of scanning, determining, performing and sending, if the data is not of a type that is likely to contain a virus (If using the technique suggested by Stang, the proxy server transmits data without performing virus detection if the data was determined not to be likely to contain a virus).

It would have been obvious to one of ordinary skill in the art at the time of the invention to employ the use of a second means for determining whether the data is of a type that is likely to contain a virus; and means for transmitting the data from the server to the destination without performing the steps of scanning, determining, performing and sending, if the data is not of a type that is likely to contain a virus in the system of Norman, as Stang teaches, so as to reduce the amount of data to be scanned for viruses and minimize delays in transmission of network traffic.

Claims 5-6 are rejected under 35 U.S.C. 103(a) as being obvious over **Norman Data Defense Systems, Inc. June 1995 (hereafter Norman)** (An Introduction to the Norman Firewall) in view of **David J. Stang, (hereafter Stang)** (ICSA's Computer Virus Handbook) and further in view of **Warner** (re: LZEXE and SCAN (PC), posting to VIRUS-L mailing list dated May 18, 1990, reprinted in VIRUS-L Digest , vol. 3, no. 99, May 21, 1990).

a) **As to claim 5,** the combination of Norman and Stang does not disclose, however Warner disclose the steps of storing the data in a temporary file at the server after the step of electronically transmitting; and wherein the step of determining includes scanning the data for a virus using the server (Warner discloses a compressed file manager which scans compressed files for viruses: "it searches the compressed file for .EXE,.COM, .OBJ, and .SYS files, then uncompresses them into a temporary file and scans that temp file" (Warner, p. 2).

It would have been obvious to the ordinary skill in the art at the time of the invention to employ the use of storing the data in a temporary file at the server after the step of electronically transmitting; and wherein the step of determining includes scanning the data for a virus using the server in the system of Norman and Stang, as Warner teaches, so as to provide a specific technique to allow files being transmitted through the network (whether compressed or uncompressed) to be checked for viruses at the network gateway before such files could do damage on destination machines.

b) **As to claim 6**, the combination of Norman, Stang and Warner discloses the step of scanning is performed using a signature scanning process (Norman states that "[a]s new viruses are discovered and analyzed, their 'signatures' are included in the virus definition file (NVC.DEF)", a file that is updated regularly (Norman, p. 9).

Claims 9-10 are rejected under 35 U.S.C. 103(a) as being obvious over **Norman Data Defense Systems, Inc. June 1995 (hereafter Norman)** (An Introduction to the Norman Firewall) in view of **David J. Stang, (hereafter Stang)** (ICSA's Computer Virus Handbook) and further in view of TIS Firewall (TIS Firewall Toolkit Overview).

a) **As to claim 9,** the combination of Norman and Stang discloses determining whether the data is being transferred into a first network by comparing the destination address to valid addresses for the first network (Norman teaches a firewall that "can identify the packets' destination" (Norman, p. 5). Moreover, conventional network security products "'read' the address information in packets and direct each to its intended destination" (Norman, p. 5). For example, a screening router applies rules that "rely on the origin and destination IP-addresses to decide if a packet is 'good' or 'bad' " (Norman, p. 3); wherein the server is a FTP proxy server (The proxy server of Norman supports proxy services for FTP (Norman, pp. 8, 11)); wherein the step of electronically receiving data comprises the steps of transferring the data from a client node to the FTP proxy server, if the data is not being transferred into the first network (Norman illustrates an FTP transaction on page 8. The two-way arrow labeled "ftp" between the proxy server inside the firewall and the external "Remote Host" shows the transfer of a file from the client node (Remote Host) to the FTP proxy server. In this case, the data is not being transferred into a first network (the external network containing the Remote Host).

The combination of Norman and Stang does not disclose, however TIS Firewall discloses wherein the step of electronically receiving data comprises the steps of transferring the data from a server task to an FTP daemon, and then from the FTP daemon to the FTP proxy server if the data is being transferred into the first network (TIS Firewall teaches a host-based application-level firewall design in which an FTP proxy controls the transfer of data files between an FTP daemon (which necessarily receives a file to be transferred from a file server) and a recipient node. A "bastion host provides application-level control" (TIS Firewall, p. 39). "The FTP application gateway is a single process that mediates FTP connections between two networks." (TIS Firewall, p. 41) "To control FTP access, the application gateway reads a configuration file, containing a list of FTP commands that should be logged, and a description of what systems are allowed to engage in FTP traffic." (TIS Firewall, pp. 41-42). Regarding proxies generally, TIS Firewall states that "[a] proxy

for a network protocol is an application that runs on a firewall host and connects specific service requests across the firewall, acting as a gateway .... Proxies can give the illusion to the software on both sides of a direct point-to-point connection. Since many proxies interpret the protocol that they manage, additional access control and audit may be performed as desired. As an example, the FTP proxy can block FTP export of files while permitting import of files, representing a granularity of control that router-based firewalls cannot presently achieve." (TIS Firewall, p. 37) Although the diagram of an application proxy on page 38 of TIS Firewall is specific to telnet rather than FTP, it shows that an application proxy is distinct from, and communicates with, an application daemon (telnetd server). TIS Firewall discloses the use of an FTP daemon ("common programs such as the FTP server, ftpd") in discussing the advantages of a proxy- based firewall design (TIS Firewall, p. 38; the WUArchive ftpd is referenced on p. 44 as an "FTP server daemon").

It would have been obvious to one of ordinary skill in the art at the time of the invention to employ the use of transferring the data from a server task to an FTP daemon, and then from the FTP daemon to the FTP proxy server if the data is being transferred into the first network in the system of Norman and Stang, as TIS Firewall discloses, so as to allow secure file transfer as well as reuse of existing FTP facilities.

b) **As to claim 10,** the combination of Norman and Stang discloses determining whether the data is being transferred into a first network by comparing the destination address to valid addresses for the first network (Norman teaches a firewall that "can identify the packets' destination" (Norman, p. 5). Moreover, conventional network security products "'read' determining whether the data is being the address information in packets and direct each to its intended destination" (Norman, p. 5). For example, a screening router applies rules that "rely on the origin and destination IP-addresses to decide if a packet is 'good' or 'bad'" (Norman, p. 3); wherein the server is a FTP proxy server (The proxy server of Norman supports proxy services for FTP (Norman, pp. 8, 11)); wherein the step of sending the data to the destination address

comprises transferring the data from the FTP proxy server to a node having the destination address, if the data is being transferred into the first network (Norman illustrates an FTP transaction on page 8. The two-way arrow labeled "ftp" between the proxy server inside the firewall and the external "Remote Host" shows the transfer of a file from the FTP proxy server to the node having the destination address (Remote Host). In this case, the data is being transferred into a first network (the external network containing the Remote Host).

The combination of Norman and Stang does not discloses, however TIS Firewall discloses wherein the step of sending the data to the destination address comprises transferring the data from the FTP proxy server to a FTP daemon, and then from an FTP daemon to a node having the destination address, if the data is not being transferred into the first network (TIS Firewall teaches a host-based application-level firewall design in which an FTP proxy controls the transfer of data files between an FTP daemon and a recipient node; the FTP daemon necessarily transmits imported files to an internal node or file server. A "bastion host provides application-level control" (TIS Firewall, p. 39). "The FTP application gateway is a single process that mediates FTP connections between two networks." (TIS Firewall, p. 41) "To control FTP access, the application gateway reads a configuration file, containing a list of FTP commands that should be logged, and a description of what systems are allowed to engage in FTP traffic." (TIS Firewall, pp. 41-42). Regarding proxies generally, TIS Firewall states that "[a] proxy for a network protocol is an application that runs on a firewall host and connects specific service requests across the firewall, acting as a gateway .... Proxies can give the illusion to the software on both sides of a direct point-to-point connection. Since many proxies interpret the protocol that they manage, additional access control and audit may be performed as desired. As an example, the FTP proxy can block FTP export of files while permitting import of files, representing a granularity of control that router-based firewalls cannot presently achieve." (TIS Firewall, p. 37). Although the diagram of an application proxy on page 38 of TIS Firewall is specific to telnet rather than FTP, it shows that an application proxy is distinct from, and communicates with, an

application daemon (telnetd server). TIS Firewall discloses the use of an FTP daemon ("common programs such as the FTP server, ftpd") in discussing the advantages of a proxy-based firewall design (TIS Firewall, p. 38; the WUArchive ftpd is referenced on p. 44 as an "FTP server daemon").

It would have been obvious to one of ordinary skill in the art at the time of the invention to employ the use sending the data to the destination address comprises transferring the data from the FTP proxy server to a FTP daemon, and then from an FTP daemon to a node having the destination address, if the data is not being transferred into the first network in the system of Norman and Stang, as TIS Firewall discloses, so as to allow secure file transfer as well as reuse of existing FTP facilities.

Claims 11-12 and 14-17 are rejected under 35 U.S.C. 103(a) as being obvious over **Norman Data Defense Systems, Inc., June 1995 (hereafter Norman)** (An Introduction to the Norman Firewall) in view of **Warner** (re: LZEXE and SCAN (PC), posting to VIRUS-L mailing list dated May 18, 1990, reprinted in VIRUS-L Digest , vol. 3, no. 99, May 21, 1990).

a) **As to claim 11,** Norman discloses a computer implemented method for detecting viruses in a mail message transferred between a first computer and a second computer (Norman teaches a firewall which, unlike a mere packet filter or a router, "combines multiple secure computing and communications devices in a single package, including a fully configured secure computer system and virus detection capability" (Norman, p. 4). The firewall "automatically checks every incoming file for viruses before letting the file through" (Norman, p. 5). Incoming files include mail messages being transferred: the firewall "has proxy services for... SMTP (e-mail)" (Norman, p. 8), the firewall runs mail forwarding software (Norman, p. 6), and the antivirus module acts on contents of electronic mail (Norman, p. 9)), the method comprising the steps of: receiving a mail

message request including a destination address (With a proxy server between an internal network and external connections, "IP packets will not pass directly from the input to the output network interfaces", because "the proxy server runs two separate connections with the proxy as the carrier in between" (Norman, p. 1). The firewall of Norman "uses nothing but proxy services to pass traffic from one network to the other. No packets will be allowed to pass directly." (Norman, p. 7.) Such a proxy server necessarily receives data transfer requests from internal network nodes. With respect to outgoing transfers, the firewall "log[s] into the workstation on the secure network to transfer the requested file" (Norman, p. 8). The firewall "can identify the packets' destination" (Norman, p. 5). Internet security products in general "'read' the address information in packets and direct each to its intended destination" (Norman, p. 5). Such devices employ packet routing and filtering, including on outgoing traffic: for example, screening router rules "rely on the origin and destination IP addresses to decide if a packet is 'good' or 'bad' "; "rules can be applied to the source and destination ports", and one can "specify separate sets of rules on incoming and outgoing connections" (Norman, p. 3)); electronically receiving the mail message at a server (Norman describes a firewall having a proxy server that receives incoming data. The proxy server stands "between the [internal] network and any external connections .... IP packets will not pass directly from the input to the output network interfaces in the proxy server environment." (Norman, p. 1.)); determining whether the mail message contains a virus, the determination of whether the mail message contains a virus comprising determining whether the mail message includes any encoded portions, decoding the encoded portions of the mail message to produced decoded portions of the mail message (Norman indicates that uuencoded files will be decoded portions for a virus, and testing whether decoded before being scanned for viruses: "Files the scanning step found any viruses; that are compressed using one of several known methods, will be uncompressed before scan. Methods currently supported include.. .UUencode." (Norman, p. 9.); performing a preset action on the mail message if the mail message contains a virus (The firewall of Norman "scans all incoming files for any of 7100+ viruses,

and sets them aside for later examination rather than forwarding them, if they are infected" (Norman, p. 5). "When a virus is located [by the firewall], the file transaction is blocked and logged." (Norman, p. 9.) The firewall "can be made to notify a network management station on the internal net through SNMP traps. If a virus.., is discovered, traps can be sent to one or several machines on the secure network." (Norman, p. 20.)); and sending the mail message to the destination address if the mail message does not contains a virus ("A network administrator may also want to control the contents of electronic mail ....Traffic that is due to be checked for viruses and hotwords are queued, and the Anti-virus/Hotword module will then scan and give clearances for each file. When a file is cleared, it is then passed on by the proxy process." (Norman, p. 9.)

Norman does not disclose, however Warner discloses storing each encoded portion of the mail message in a separate temporary file and scanning each of the decoded portions for a virus, and testing whether the scanning step found any viruses (Warner discloses a compressed file manager which scans compressed files for viruses: "it searches the compressed file for .EXE, .COM, .OBJ, and .SYS files, then uncompresses them into a temporary file and scans that temp file" (Warner, p. 2)).

It would have been obvious to one of ordinary skill in the art at the time of the invention to employ the use of storing each encoded portion of the mail message in a separate temporary file and scanning each of the decoded portions for a virus, and testing whether the scanning step found any viruses in the system of Norman, as Warner teaches, so as to allow compressed files being transmitted through the network to be checked for viruses at the network gateway before such files could do damage on destination machines.

b) **As to claim 12**, the combination of Norman and Warner discloses the step of determining whether the mail message includes any encoded portions searches

for uuencoded portions ("Files that are compressed using of several known methods, will be uncompressed before scan: Methods currently supported include...UUencode." (Norman, p. 9.)).

c) **As to claim 14**, the combination of Norman and Warner discloses wherein the step of determining whether the mail message contains a virus, further comprises the steps of: storing the message in a temporary file; scanning the temporary file for viruses; and testing whether the scanning step found a virus (Warner discloses a compressed file manager which scans compressed files for viruses: "it searches the compressed file for .EXE, .COM, .OBJ, and .SYS files, then uncompresses them into a temporary file and scans that temp file" (Warner, p. 2)).

d) **As to claim 15**, the combination of Norman and Warner discloses wherein step of scanning is performed using a signature scanning process (Norman states that "[a]s new viruses are discovered and analyzed, their 'signatures' are included in the virus definition file (NVC.DEF)", a file that is updated regularly (Norman, p. 9)).

e) **As to claim 16**, the combination of Norman and Warner discloses wherein the step of performing a preset action on the mail message comprises performing one step from the group of: transferring the mail message unchanged (Transmitting data unchanged, even if it contains a virus, simply represents the ordinary operation of prior art network gateways which performed no antivirus scanning); not transferring the mail message (The firewall of Norman "scans all incoming files for any of 7100+ viruses, and sets them aside for later examination rather than forwarding them, if they are infected" (Norman, p. 5). "When a virus is located [by the firewall], the file transaction is blocked and logged." (Norman, p. 9.); storing the mail message as a file with a new name and notifying a recipient of the mail message request of the new file name ("The [firewall] system can even be configured to record the contents of packets and to store suspect packets for later review by a security officer." (Norman, p. 5.) At the time

the invention was made a person having ordinary skill in the art would have readily appreciated that stored packet contents could be given a unique file name, and that the firewall system could notify the recipient of the file name to allow the recipient to request access to the file); **and** creating a modified mail message by writing the output of the determining step into the modified mail message and transferring the mail message to the destination address.

f) **As to claim 17**, the combination of Norman and Warner discloses wherein the step of performing a preset action on the mail message comprises performing one step from the group of: transferring the mail message unchanged (Transmitting data unchanged, even if it contains a virus, simply represents the ordinary operation of prior art network gateways which performed no antivirus scanning); transferring the mail message with the encoded portions having a virus deleted (According to the recitation in claim 11, encoded portion is stored in a separate temporary file, decoded, and scanned for viruses. At the time the invention was made a person having ordinary skill in the art would have found it rudimentary to configure the firewall system of Norman to delete a particular infected portion from the original mail message (since its precise location within the original mail message file would be known) and to transmit the modified mail message using ordinary electronic mail techniques); renaming the encode portions of the mail message containing a virus, and storing the renamed portions as files in a specified directory on the server and notifying a recipient of the renamed files and directory (At the time the invention was made a person having ordinary skill in the art would have readily appreciated that the temporary file recited in claim 11 must have a known path name indicating its location in some directory in the file system. It would have been rudimentary to copy such a file to a specified file system directory using basic operating system facilities, the copy having a new path name. Moreover, it would have been obvious at the time the invention was made to a person having ordinary skill in the art to configure the firewall system to notify the recipient of the path name of the file,

using known electronic mail techniques, because it would enable the recipient to request access to the file); and writing the output of the determining step into the mail message in place of respective encoded portions that contain a virus to create a modified mail message and sending the modified mail message (Modification by the mail forwarding system of the data in a mail message to include the output of a particular process simply uses file modification and electronic mail techniques well known in the art at the time the invention was made. It would have been obvious at the time the invention was made to a person having ordinary skill in the art to modify the firewall system of Norman by having the system edit a mail message that has had infected encoded portions removed to contain the result of the scanning process in the message, and then having the system send the message to the destination, because it would allow the recipient to know that a particular sender had sent infected data).

CORRESPONDENCE

All correspondence relating to this ex parte reexamination proceeding should be directed:

By EFS: Registered users may submit via the electronic filing system EFS-Web, at https://sportal.uspto.gov/authenticate/authenticateuserlocalepf.html.

By Mail to: Mail Stop *Ex Parte* Reexam
Central Reexamination Unit
Commissioner for Patents
United States Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

By FAX to: (571) 273-9900
Central Reexamination Unit

By hand: Customer Service Window
Randolph Building
401 Dulany Street
Alexandria, VA 22314

For EFS-Web transmissions, 37 CFR 1.8(a)(1)(i) (C) and (ii) states that correspondence (except for a request for reexamination and a corrected or replacement request for reexamination) will be considered timely filed if (a) it is transmitted via the Office's electronic filing system in accordance with 37 CFR 1.6(a)(4), and (b) includes a certificate of transmission for each piece of correspondence stating the date of transmission, which is prior to the expiration of the set period of time in the Office action.

Any inquiry concerning this communication or earlier communications from the Examiner, or as to the status of this proceeding, should be directed to the Central Reexamination Unit at telephone number (571) 272-7705.

Signed:

/Minh Dieu Nguyen/

Minh Dieu Nguyen
Primary Examiner
USPTO, Art Unit 3992
(571) 272-3873

Conferee: /ALN/

Conferee: /ESK/