1915 class to store the SMTP Server, the from address, and the recipient address to be used in sending configuration or status information to the central monitoring workstation 945, respectively, in the system registry of the remote monitoring workstation 911.

[0099] Figure 24 is a class diagram illustrating one embodiment of the ODBC Interface 1109 module of Figure 11 according to the present invention. The ODBC Interface 1109 module is responsible for interfacing with the database 913 that maintains the information pertaining to the SNMP devices being monitored by a particular remote monitoring workstation 911. In this embodiment, the database is registered as an ODBC database, and, therefore, the database 913 has available the appropriate supporting ODBC drivers. The ODBC Interface 1109 module includes five classes: CSendODBCInterface 2401, CDeviceInformationData 2403, CDeviceDatabase 2407, CDevicePersistentStatus 2411, and CDevicePerDatabase 2413. The CSendODBCInterface 2401 class provides the interface through which the functionality supported by the ODBC Interface 1109 module is accessed.

[0100] The CDevicemformationData 2403 class provides methods for obtaining and storing configuration information of the monitored devices in the database 913. The CDeviceDatabase 2407 class provides an interface between the CDeviceInformationData 2403 class and the actual database 913 that contains the configuration information. The CDeviceInformationData 2403 class uses the DeviceInfo structure to store the configuration information in the database 913.

[0101] The CDevicePersistentStatus 2411 class provides methods for obtaining and storing the first type of information of the monitored devices in the database. The CDevicePerDatabase 2407 provides an interface between the CDevicePersistentStatus 2411 class and the actual database that contains the first type of information. The CDevicePersistentStatus 2411 class uses the DevicePerStatus structure to store the first type of information in the database 913.

[0102] Both the CDeviceDatabase 2407 class and the CDevicePerDatabase 2413 class are derived from the CRecordset 2417 class available in the Microsoft Foundation Classes (MFC).

[0103] Obviously, numerous modifications and variations of the present invention are possible in light of the above teachings. It is therefore to be understood that within the scope of the appended claims, the invention may be practiced otherwise than as specifically described herein.

Claims

1. A method for transferring monitoring information from a remote monitor to a central monitor, comprising the steps of:

   creating a communication link by the remote monitor between the remote monitor and a message transfer server;
   formatting the monitoring information into a standard format;
   sending the monitoring information in the standard format as an attachment to a message from the remote monitor to the message transfer server via the communication link using a transfer protocol;
   sending the message from the message transfer server to the central monitor over a network using the transfer protocol; and
   closing the communication link by the remote monitor, wherein
   the monitoring information includes at least one of configuration information and status information corresponding to at least one networked device monitored by the remote monitor.

2. The method of Claim 1, further comprising the steps of:

   storing a source address corresponding to the remote monitor in a first memory of the remote monitor;
   storing a destination address corresponding to the central monitor in a second memory of the remote monitor; and
   querying the first memory for the source address by the remote monitor;
   querying the second memory for the destination address by the remote monitor;
   creating a message header including the source address and the destination address by the remote monitor, wherein
   the message includes the message header.

3. The method of Claim 2, wherein at least one of the first memory and the second memory comprises a system registry of the remote monitor.

4. The method of Claim 1, wherein the standard format does not change with a type of monitoring information being

ForcePoint LLC
EX-1009
Page154

formatted.

5. The method of Claim 1, wherein the at least one networked device comprises at least one of a printer, a facsimile device, a copier, a scanner, an appliance, a video cassette recorder, a digital camera, a cellular phone, and a personal data assistant.

6. The method of Claim 1, further comprising the steps of:

gathering the monitoring information by the remote monitor from the at least one network device; and
storing the monitoring information gathered in the gathering step by the remote monitor in a digital repository coupled to the remote monitor.

7. The method of Claim 6, wherein the gathering step comprises gathering the monitoring information using a simple network management protocol.

8. The method of Claim 1, wherein the transfer protocol comprises a store-and-forward protocol.

9. The method of Claim 1, wherein the transfer protocol comprises a simple mail transfer protocol.

10. The method of Claim 1, wherein the transfer protocol comprises a direct connection protocol.

11. The method of Claim 10, wherein the direct connection protocol comprises at least one of a file transfer protocol and a hyper text transfer protocol.

12. The method of Claim 1, wherein:

the message comprises an Internet e-mail message, and
the attachment comprises a multipurpose Internet mail extensions attachment to the Internet e-mail message.

13. The method of Claim 1, wherein the configuration information comprises at least one of an Internet protocol address and a media access control address of the at least one networked device.

14. The method of Claim 1, wherein the status information comprises at least one of an alarm and an operational status of the at least one networked device.

15. The method of Claim 1, further comprising the step of:

encrypting the monitoring information by the remote monitor prior to the sending the monitoring information step.

16. The method of Claim 15, further comprising the step of:

encoding the monitoring information by the remote monitor after the encrypting step and prior to the sending the monitoring information step.

17. The method of Claim 16, wherein the encoding step comprises encoding the monitoring information using a base 64 encoding.

18. The method of Claim 1, further comprising the step of:

encoding the monitoring information by the remote monitor prior to the sending the monitoring information step.

19. A computer-implemented system for transferring monitoring information from a remote monitor to a central monitor, comprising:

a processor; and
a computer readable medium encoded with processor readable instructions that when executed by the processor implement

ForcePoint LLC
EX-1009
Page155

a communication link mechanism configured to create a communication link between the remote monitor and a message transfer server and to close the communication link after monitoring information has been sent by the remote monitor to the message transfer server, the monitoring information including at least one of configuration information and status information corresponding to at least one network device monitored by the remote monitor,

a monitoring information formatting mechanism configured to format the monitoring information into a standard format, and

a sending mechanism configured to send the monitoring information to the message transfer server as an attachment to a message via the communication link using a transfer protocol.

20. The system of Claim 19, wherein the computer readable medium is further encoded with processor readable instructions that when executed by the processor further implement

an encrypting mechanism configured to encrypt the monitoring information prior to sending the monitoring information to the message transfer server.

21. The system of Claim 20, wherein the computer readable medium is further encoded with processor readable instructions that when executed by the processor further implement

an encoding mechanism configured to encode the monitoring information after the encrypting mechanism encrypts the monitoring information and prior to sending the monitoring information to the message transfer server.

22. The system of Claim 21, wherein the encoding mechanism is further configured to encode the monitoring information using a base 64 encoding.

23. The system of Claim 19, wherein the computer readable medium is further encoded with processor readable instructions that when executed by the processor further implement

an encoding mechanism configured to encode the monitoring information prior to sending the monitoring information to the message transfer server.

24. The system of Claim 19, wherein the computer readable medium is further encoded with processor readable instructions that when executed by the processor further implement

a message address mechanism configured to

store a source address corresponding to the remote monitor in a first memory of the remote monitor, and

store a destination address corresponding to the central monitor in a second memory of the remote monitor; and

a message header manipulation mechanism configured to

query the first memory for the source address,

query the second memory for the destination address, and

create a header including the source address and the destination address,

wherein

the message includes the header.

25. The system of Claim 24, wherein at least one of the first memory and the second memory comprise a system registry of the remote monitor.

26. The system of Claim 19, wherein the transfer protocol comprises a store-and-forward protocol.

27. The system of Claim 19, wherein the transfer protocol comprises a simple mail transfer protocol.

28. The system of Claim 19, wherein:

the message comprises an Internet e-mail message, and

the attachment comprises a multipurpose Internet mail extensions attachment to the Internet e-mail message.

29. The system of Claim 19, wherein the standard format does not change with a type of monitoring information being formatted.

30. The system of Claim 19, wherein the at least one network device comprises at least one of a printer, a facsimile device, a copier, a scanner, an appliance, a video cassette recorder, a digital camera, a cellular phone, and a

ForcePoint LLC
EX-1009
Page156

personal data assistant.

31. The system of Claim 19, wherein the computer readable medium is further encoded with processor readable instructions that when executed by the processor further implement

an database management mechanism configured to store the monitoring information in a digital repository coupled to the remote monitor and retrieve the monitoring information from the digital repository.

32. The system of Claim 19, wherein the transfer protocol comprises a direct connection protocol.

33. The system of Claim 19, wherein the transfer protocol comprises at least one of a file transfer protocol and a hyper text transfer protocol.

34. The system of Claim 19, wherein the configuration information comprises at least one of an Internet protocol address and a media access control address of the at least one network device.

35. The system of Claim 19, wherein the status information comprises at least one of an alarm and an operational status of the at least one network device.

36. A computer program product, comprising:

a computer storage medium; and

a computer program code mechanism embedded in the computer storage medium for causing a computer to transfer monitoring information of at least one network device from a remote monitor to a central monitor, the computer program code mechanism having

a first computer code device configured to create a communication link between the remote monitor and a message transfer server and to close the communication link after monitoring information has been sent by the remote monitor to the message transfer server, the monitoring information including at least one of configuration information and status information corresponding to at least one network device monitored by the remote monitor,

a second computer code device configured to format the monitoring information into a standard format, and

a third computer code device configured to send the monitoring information to the message transfer server as an attachment to a message via the communication link using a transfer protocol.

37. The computer program product of Claim 36, wherein the computer program code mechanism further having

a fourth computer code device configured to encrypt the monitoring information prior to sending the monitoring information to the message transfer server.

38. The computer program product of Claim 37, wherein the computer program code mechanism further having

a fifth computer code device configured to encode the monitoring information after the fourth computer code device encrypts the monitoring information and prior to sending the monitoring information to the message transfer server.

39. The computer program product of Claim 38, wherein the fifth computer code device is further configured to encode the monitoring information using a base 64 encoding.

40. The computer program product of Claim 36, wherein the computer program code mechanism further having

a fourth computer code device configured to encode the monitoring information prior to sending the monitoring information to the message transfer server.

41. The computer program product of Claim 36, wherein the computer program code mechanism further having

a fourth computer code device configured to

store a source address corresponding to the remote monitor in a first memory of the remote monitor, and

store a destination address corresponding to the central monitor in a second memory of the remote monitor; and

a fifth computer code device configured to

query the first memory for the source address,

query the second memory for the destination address, and

ForcePoint LLC
EX-1009
Page157

create a header including the source address and the destination address,

wherein

the message includes the header.

42. The computer program product of Claim 41, wherein at least one of the first memory and the second memory comprise a system registry of the remote monitor.

43. The computer program product of Claim 36, wherein the transfer protocol comprises a store-and-forward protocol.

44. The computer program product of Claim 36, wherein the transfer protocol comprises a simple mail transfer protocol.

45. The computer program product of Claim 36, wherein:

the message comprises an internet e-mail message, and

the attachment comprises a multipurpose Internet mail extensions attachment to the Internet e-mail message.

46. The computer program product of Claim 36, wherein the standard format does not change with a type of monitoring information being formatted.

47. The computer program product of Claim 36, wherein the at least one network device comprises at least one of a printer, a facsimile device, a copier, a scanner, an appliance, a video cassette recorder, a digital camera, a cellular phone, and a personal data assistant.

48. The computer program product of Claim 36, wherein the computer program code mechanism further having

a fourth computer code device configured to store the monitoring information in a digital repository coupled to the remote monitor and retrieve the monitoring information from the digital repository.

49. The computer program product of Claim 36, wherein the transfer protocol comprises a direct connection protocol.

50. The computer program product of Claim 36, wherein the transfer protocol comprises at least one of a file transfer protocol and a hyper text transfer protocol.

51. The computer program product of Claim 36, wherein the configuration information comprises at least one of an Internet protocol address and a media access control address of the at least one network device.

52. The computer program product of Claim 36, wherein the status information comprises at least one of an alarm and an operational status of the at least one network device.

53. The computer program product of Claim 37, wherein a new encryption technique may be added by modifying the fourth computer code device without modifying the first computer code device, the second computer code device, the third computer code device, or an application that interfaces with the computer program product.

ForcePoint LLC
EX-1009
Page158

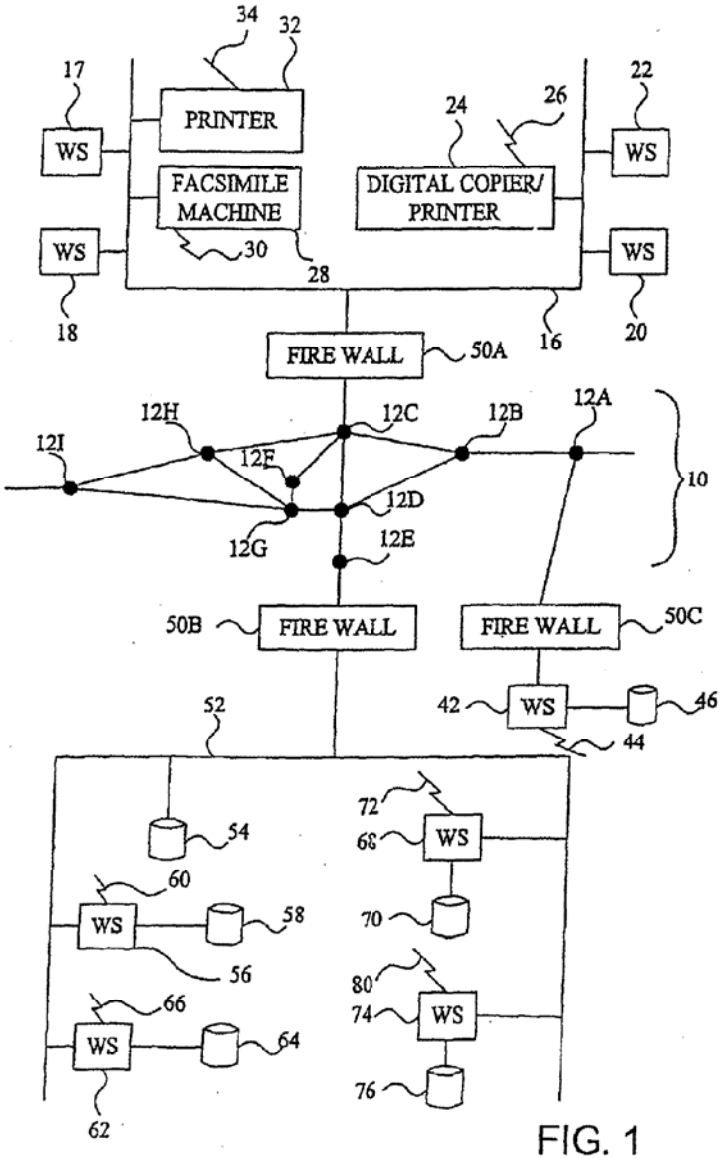


FIG. 1

ForcePoint LLC
EX-1009
Page159

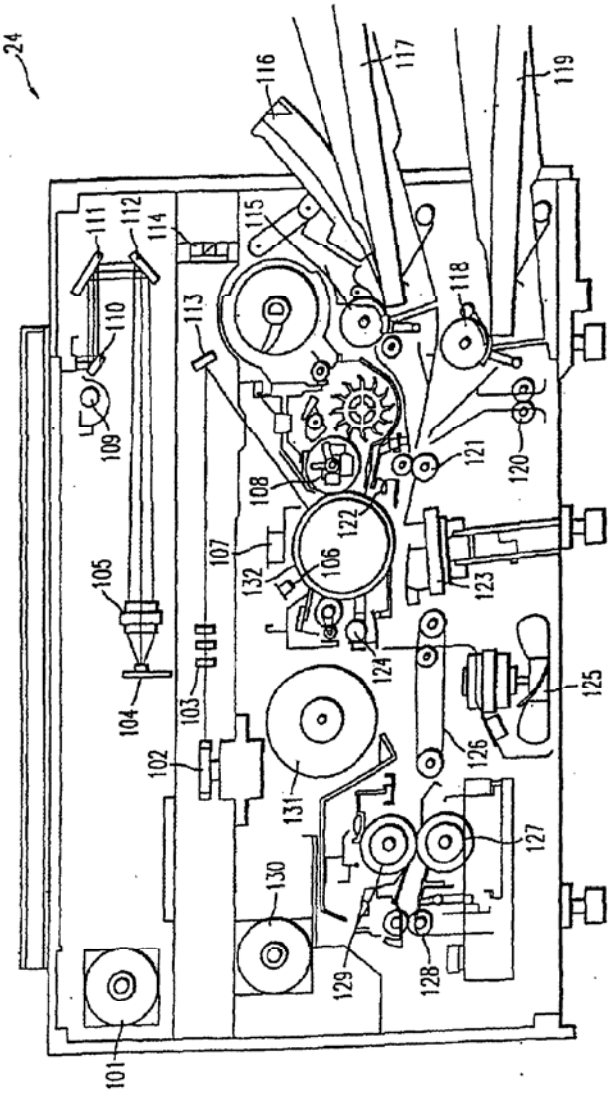


FIG. 2

ForcePoint LLC
EX-1009
Page160

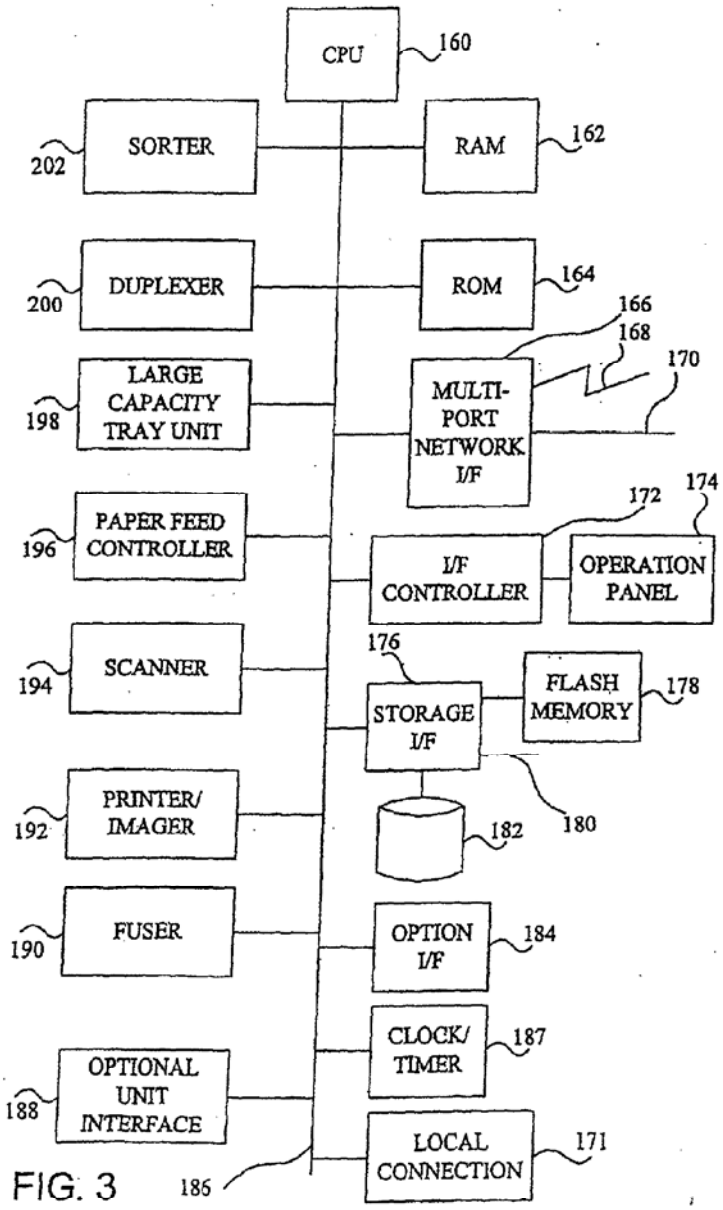


FIG. 3

ForcePoint LLC
EX-1009
Page161

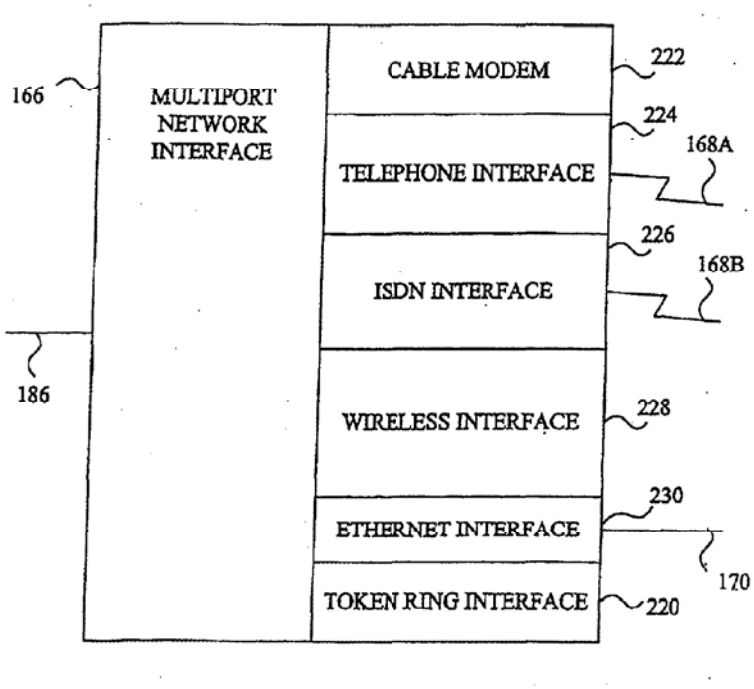


FIG. 4

ForcePoint LLC
EX-1009
Page162

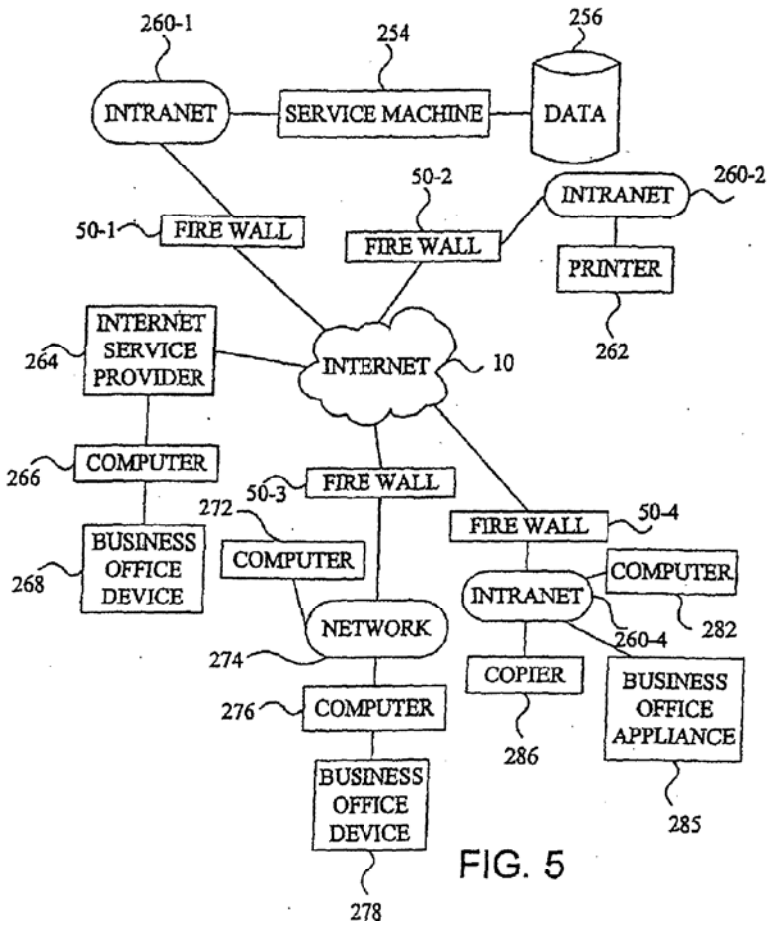


FIG. 5

ForcePoint LLC
EX-1009
Page163

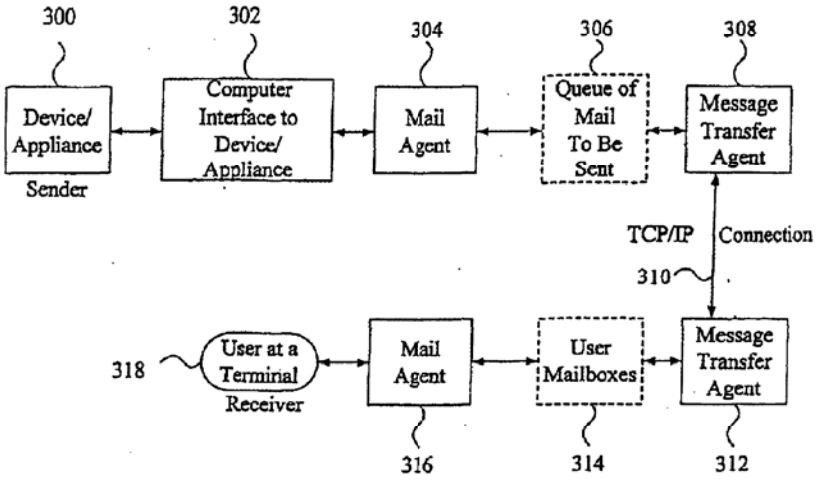


FIG. 6A

ForcePoint LLC
EX-1009
Page164

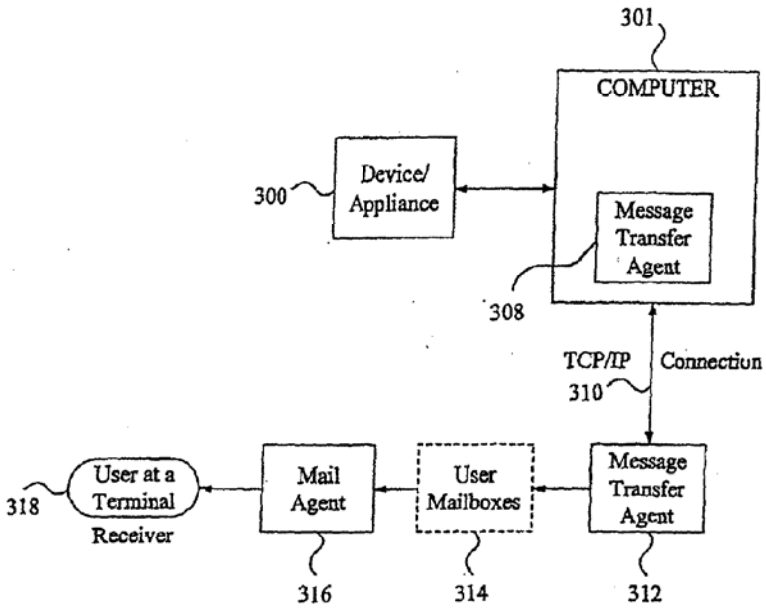


FIG. 6B

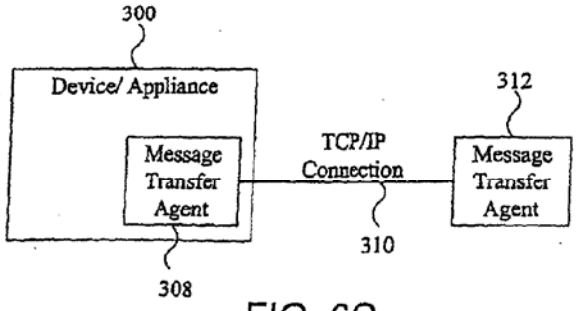


FIG. 6C

ForcePoint LLC
EX-1009
Page165

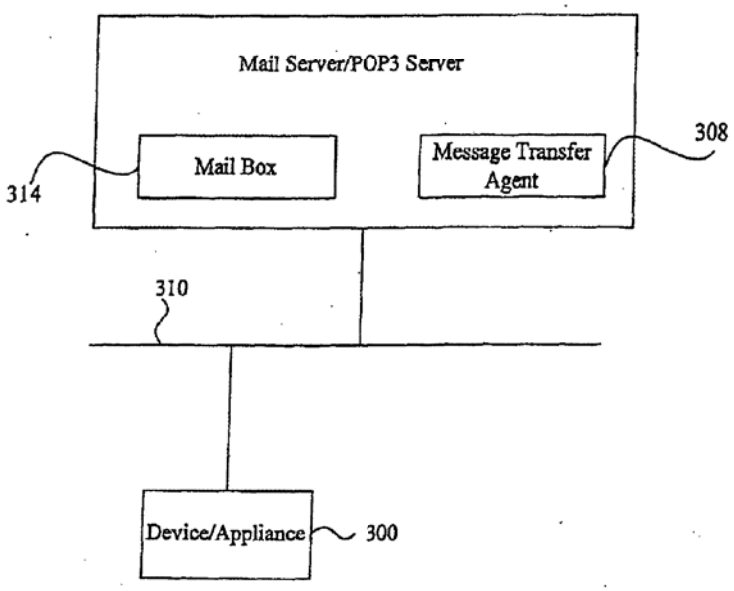


Fig. 6D

ForcePoint LLC
EX-1009
Page166

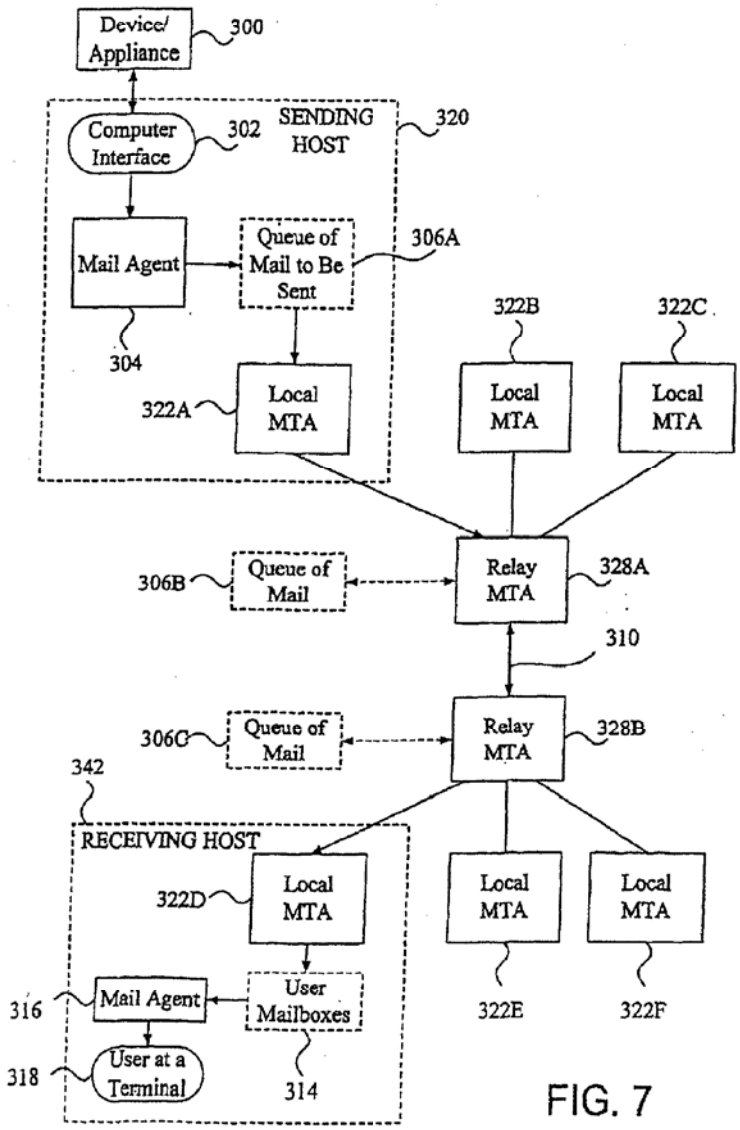


FIG. 7

ForcePoint LLC
EX-1009
Page167

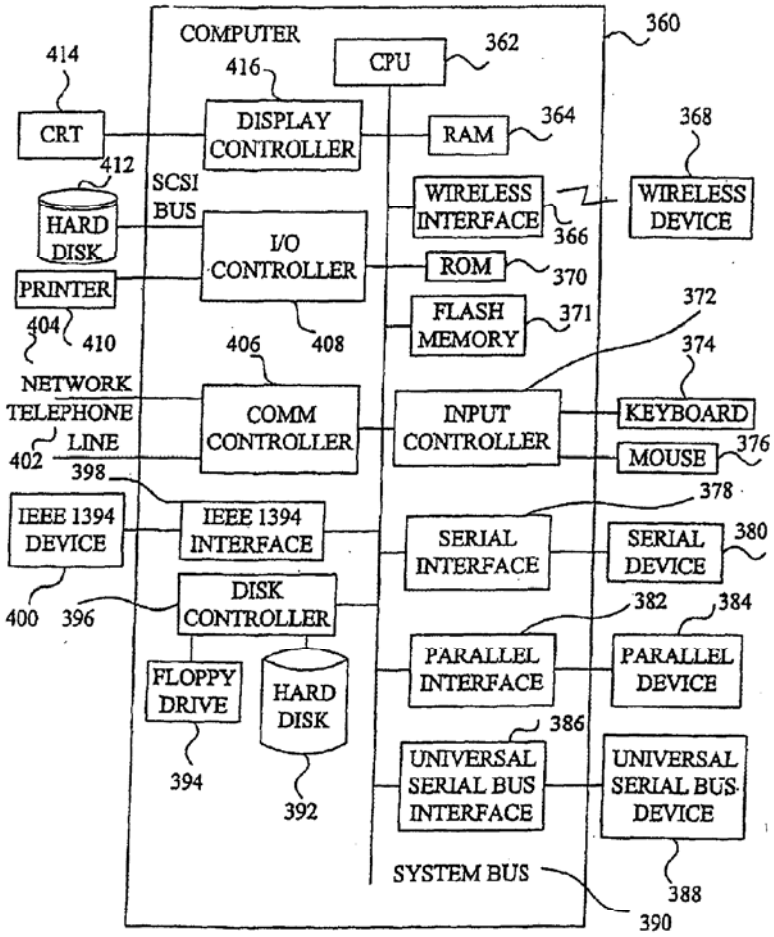


FIG. 8

ForcePoint LLC
EX-1009
Page168

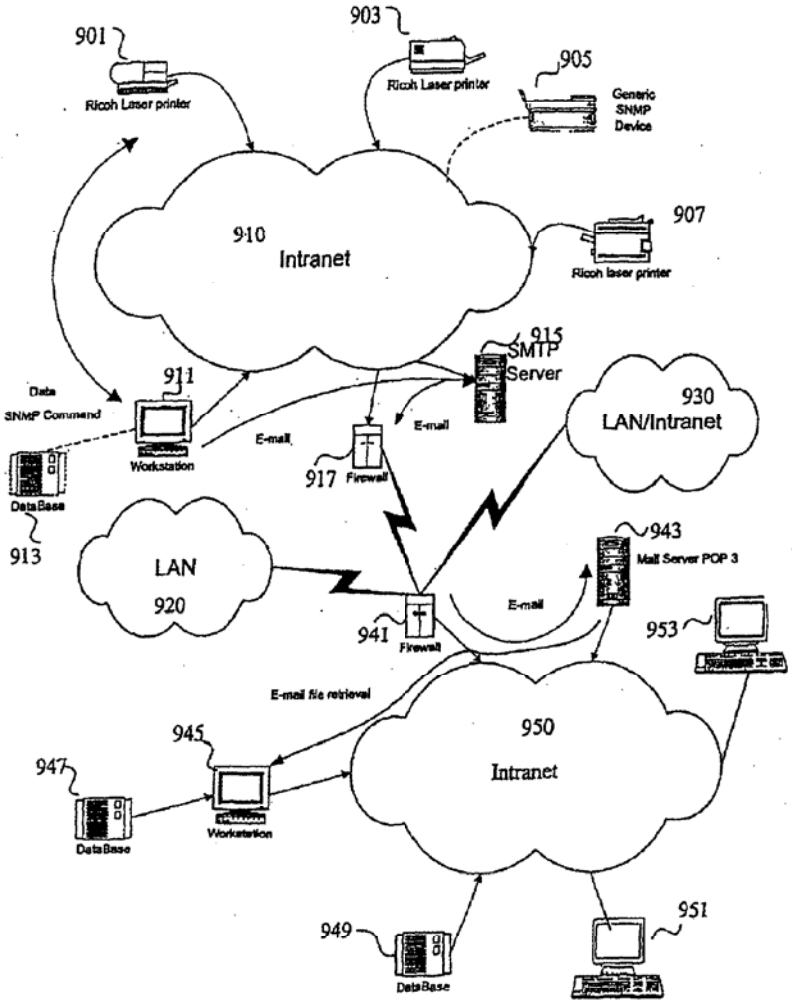


Figure 9

ForcePoint LLC
EX-1009
Page169

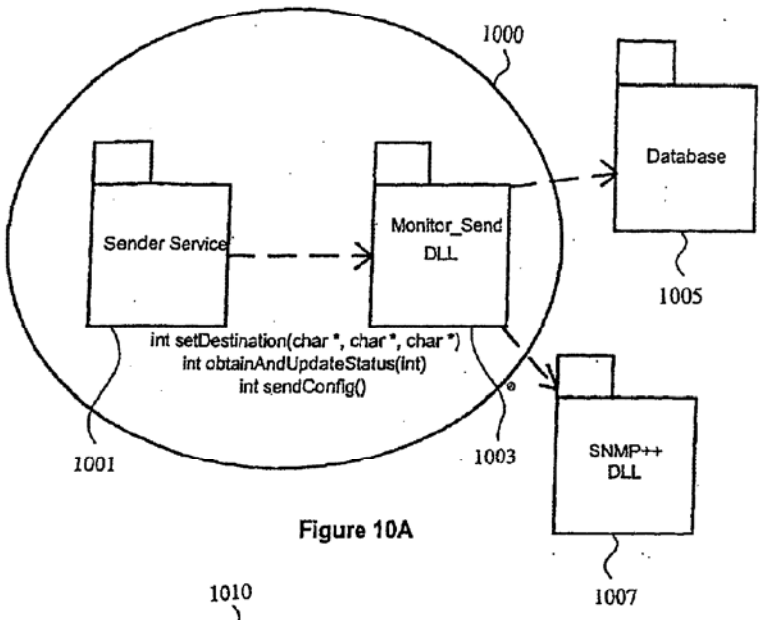


Figure 10A

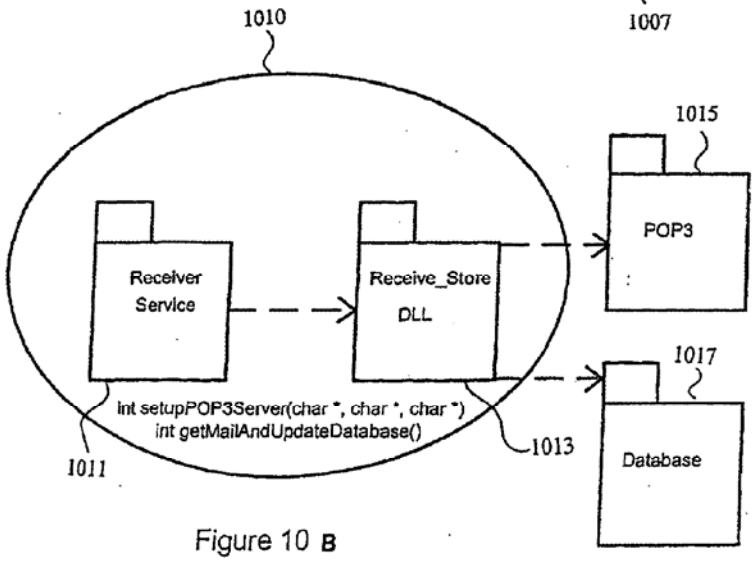


Figure 10 B

ForcePoint LLC
EX-1009
Page170

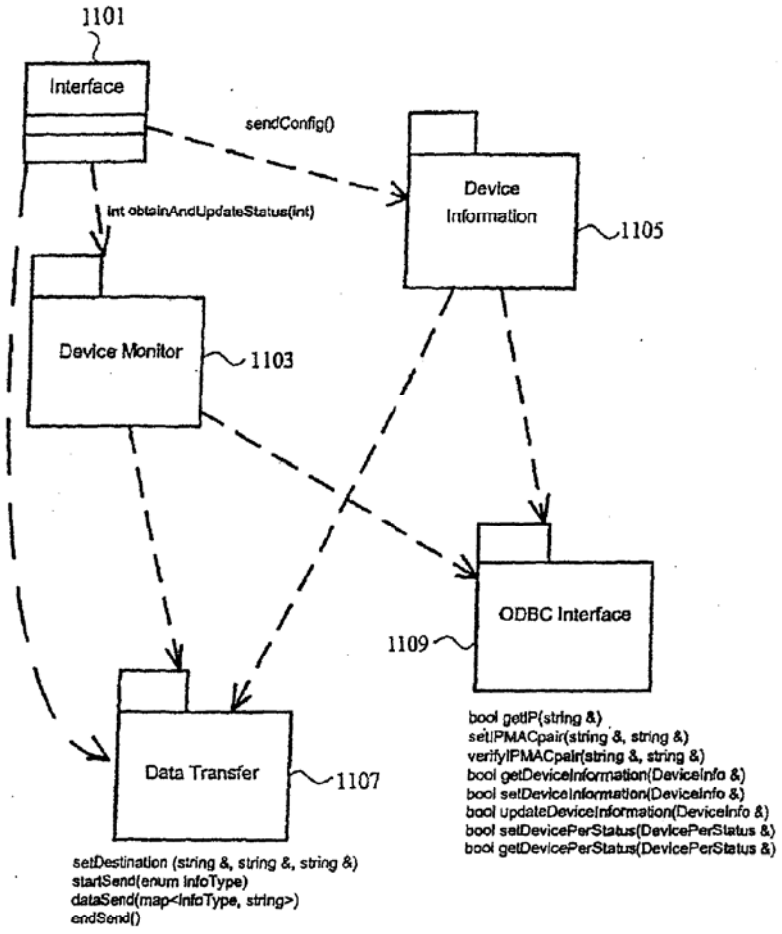


Figure 11

ForcePoint LLC
EX-1009
Page171

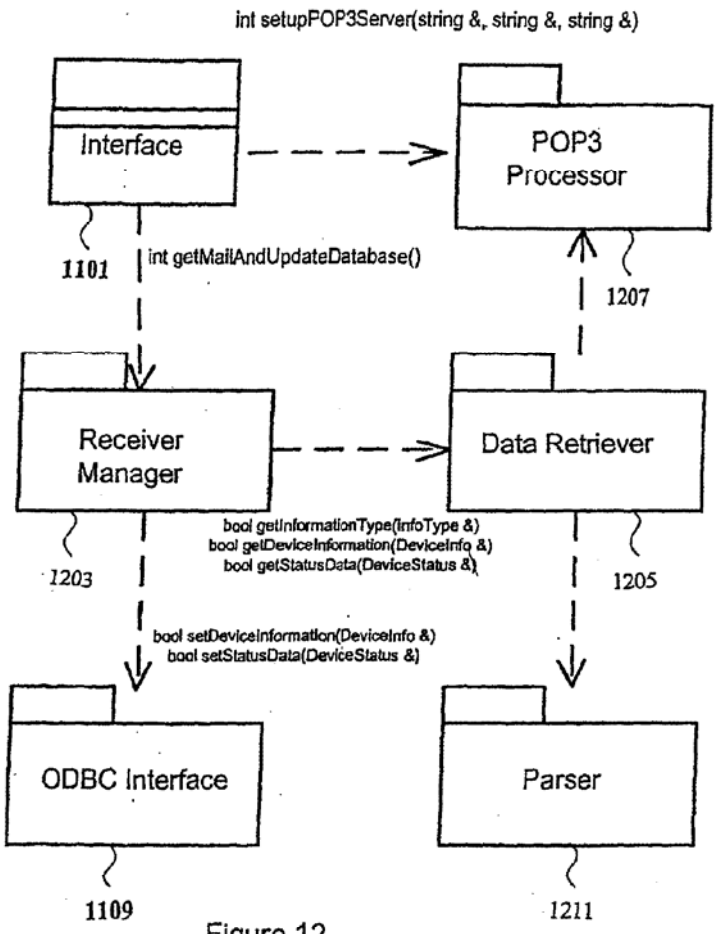
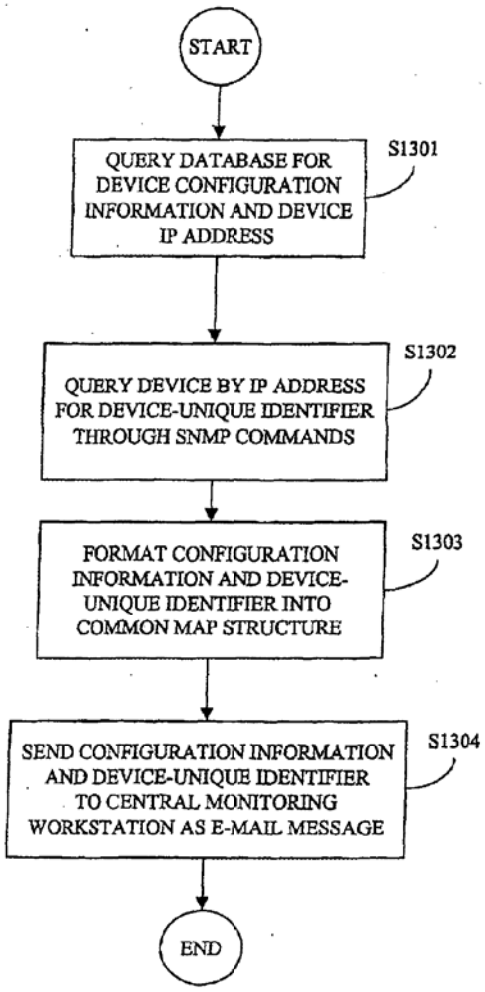


Figure 12

ForcePoint LLC
EX-1009
Page172

FIGURE 13A



ForcePoint LLC
EX-1009
Page173

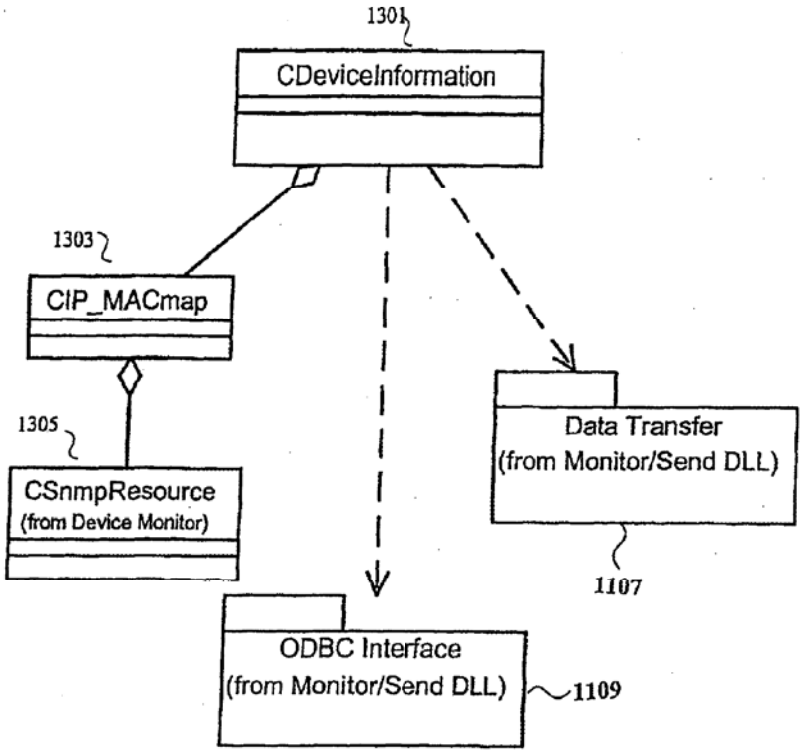


Figure 13B

ForcePoint LLC
EX-1009
Page174

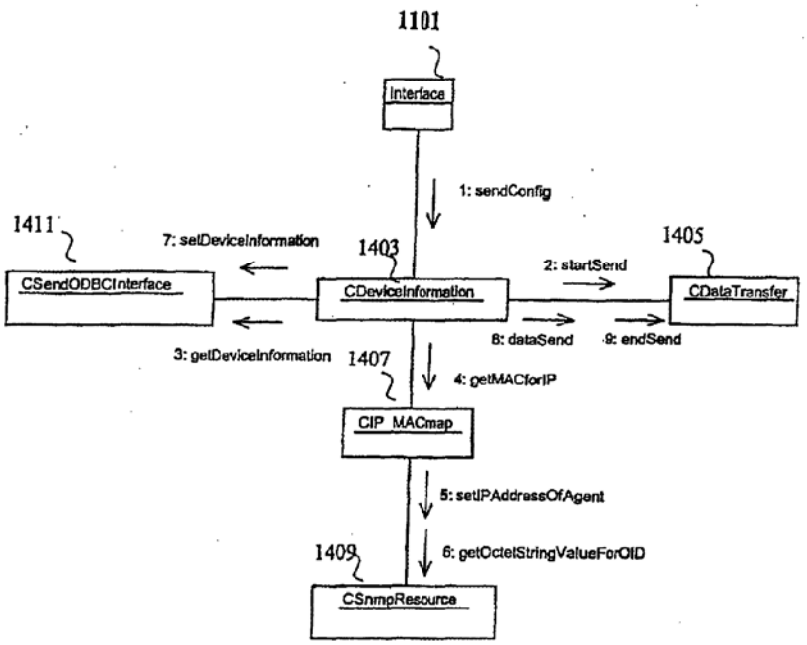
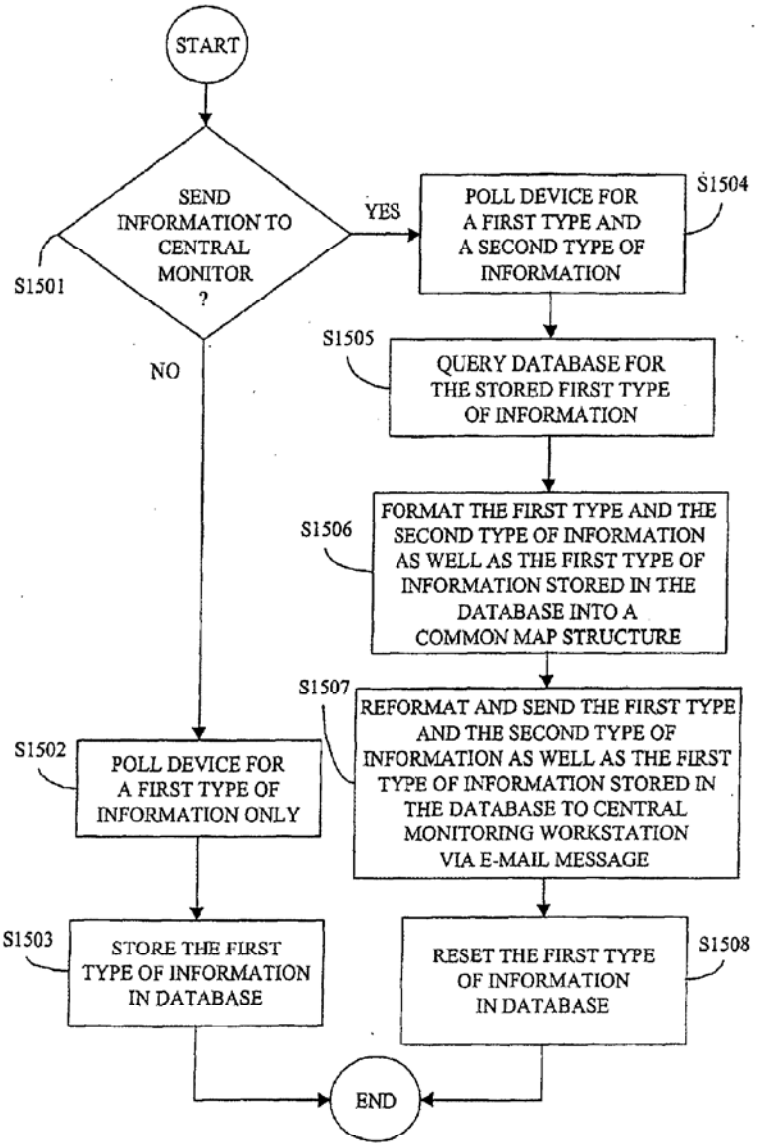


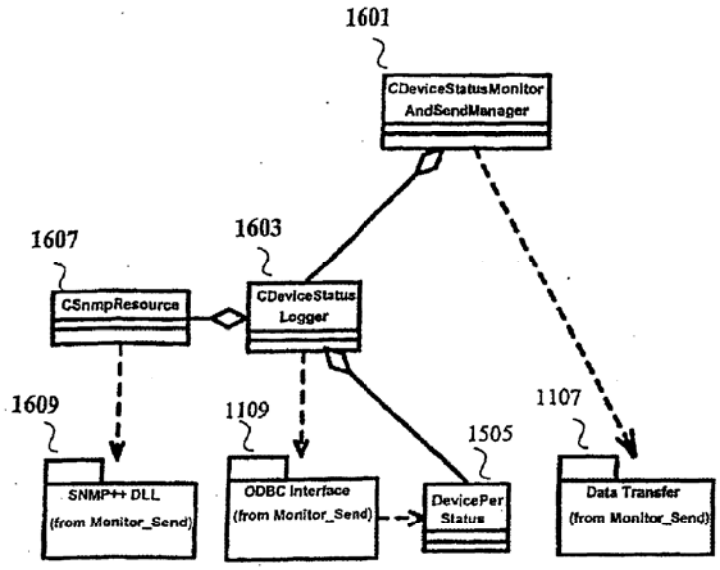
Figure 14

ForcePoint LLC
EX-1009
Page175

FIGURE 15A



ForcePoint LLC
EX-1009
Page176



Device Monitor Package Class Structure

Figure 15 B

ForcePoint LLC
EX-1009
Page177

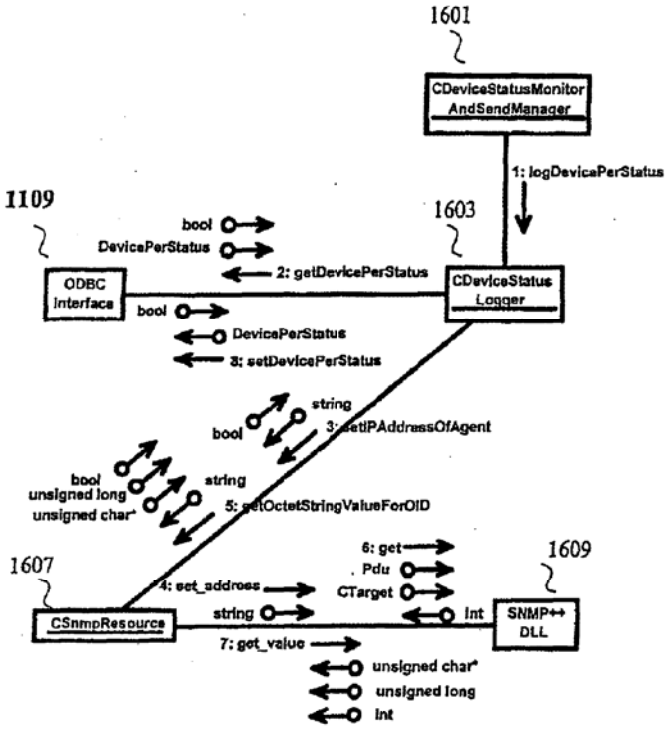


Figure 16

ForcePoint LLC
EX-1009
Page178

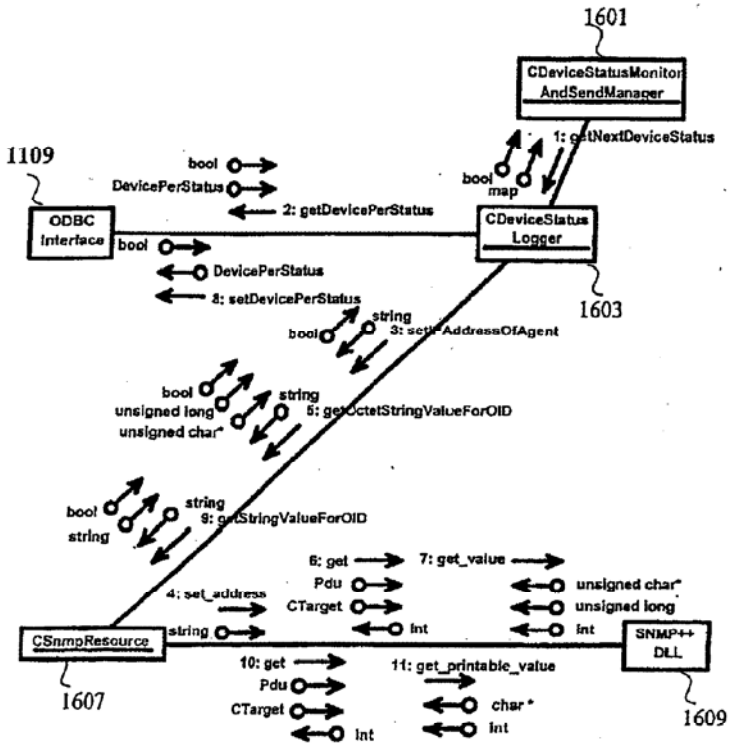


Figure 17

ForcePoint LLC
EX-1009
Page179

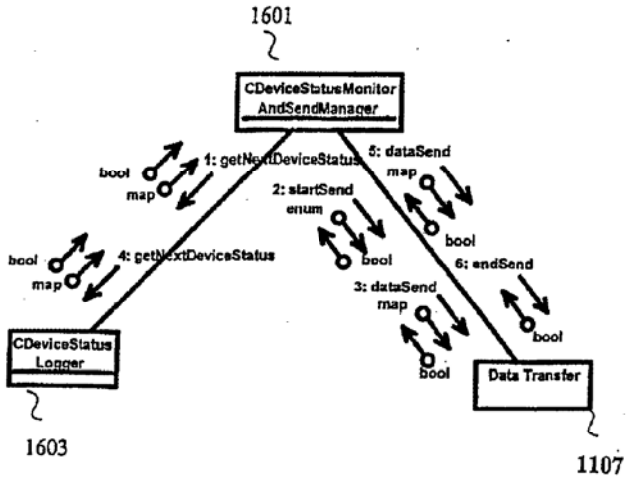
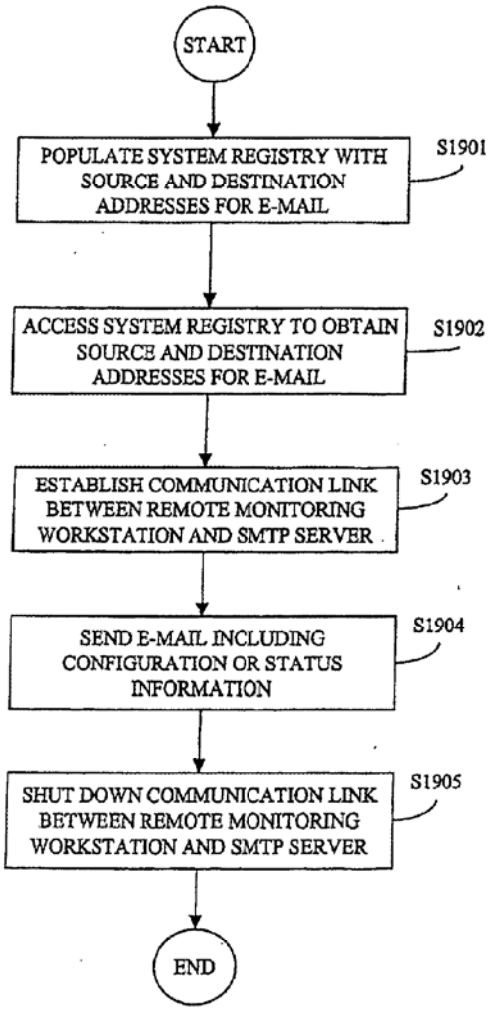


Figure 18

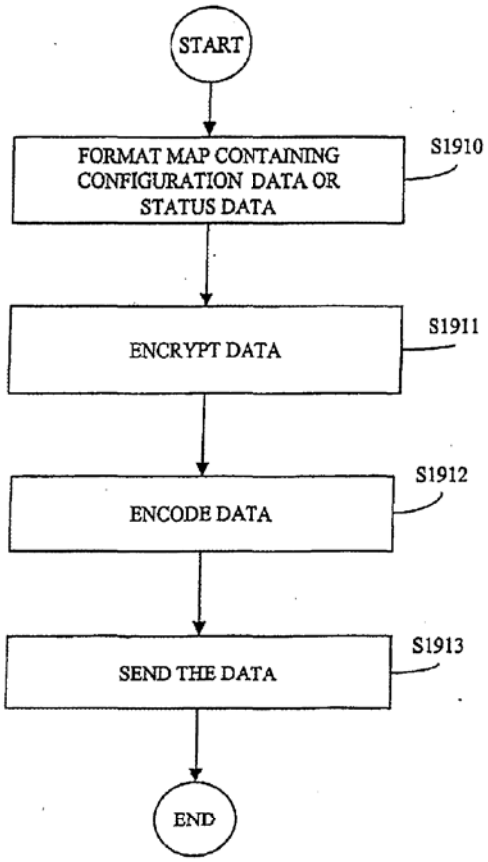
ForcePoint LLC
EX-1009
Page180

FIGURE 19A

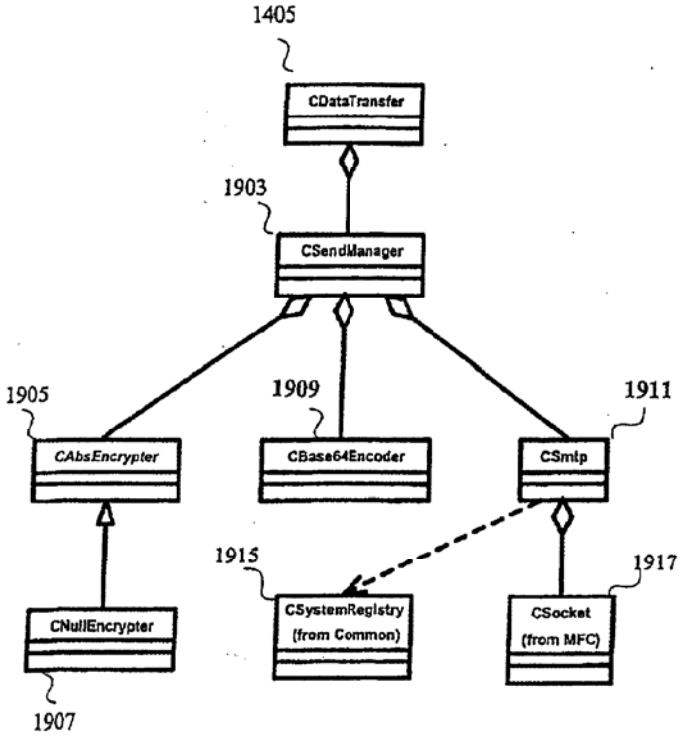


ForcePoint LLC
EX-1009
Page181

FIGURE 19B



ForcePoint LLC
EX-1009
Page182



Data Transfer Package

Figure 19C

ForcePoint LLC
EX-1009
Page183

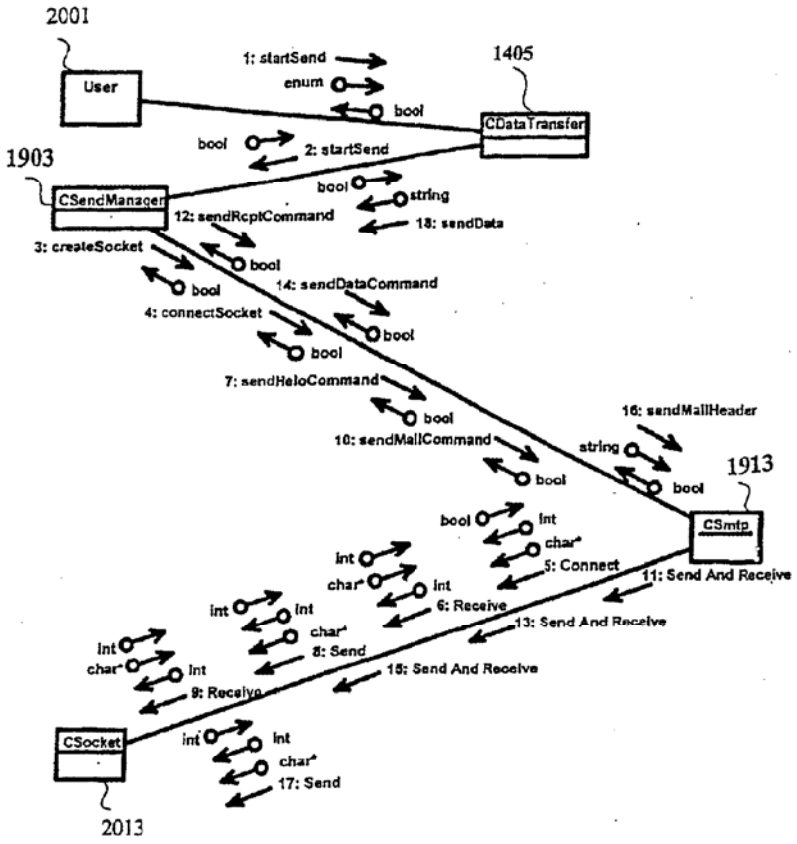


Figure 20 A

ForcePoint LLC
EX-1009
Page184

```
DataType, DeviceInformation
&&&&&
uniqueID, 00 00 aa 5e 1d fc
dateTime, 2000:08:14:14:57:15
Manufacturer, Xerox
Model, DocuPrint N4025 Network Laser Printer
...
PhoneNumber, 408-954-5445
&&&&&
uniqueID, 08 00 11 0f a5 33
dateTime, 2000:08:14:14:57:15
Manufacturer, Tektronix Inc
Model, Phaser 750P
...
PhoneNumber, 408-954-5429
#####
```

Figure 20B

ForcePoint LLC
EX-1009
Page185

```
DataType, StatusInformation
&&&&&
uniqueID, 00 00 aa 5e 1d fc
dateTime, 2000:08:14:14:57:15
errorCount, 0
lowPaper, 0
noPaper, 0
lowToner, 0
noToner, 0
doorOpen, 0
jammed, 0
offline, 0
serviceRequested, 0
configChangeCount, 14
pageLifeCount, 1025
&&&&&
uniqueID, 08 00 11 0f a5 33
dateTime, 2000:08:14:14:57:15
errorCount, 0
lowPaper, 0
noPaper, 0
lowToner, 0
noToner, 0
doorOpen, 0
jammed, 0
offline, 0
serviceRequested,
configChangeCount, 1
paperLifeCount, 202
#####
```

Figure 20C

ForcePoint LLC
EX-1009
Page186

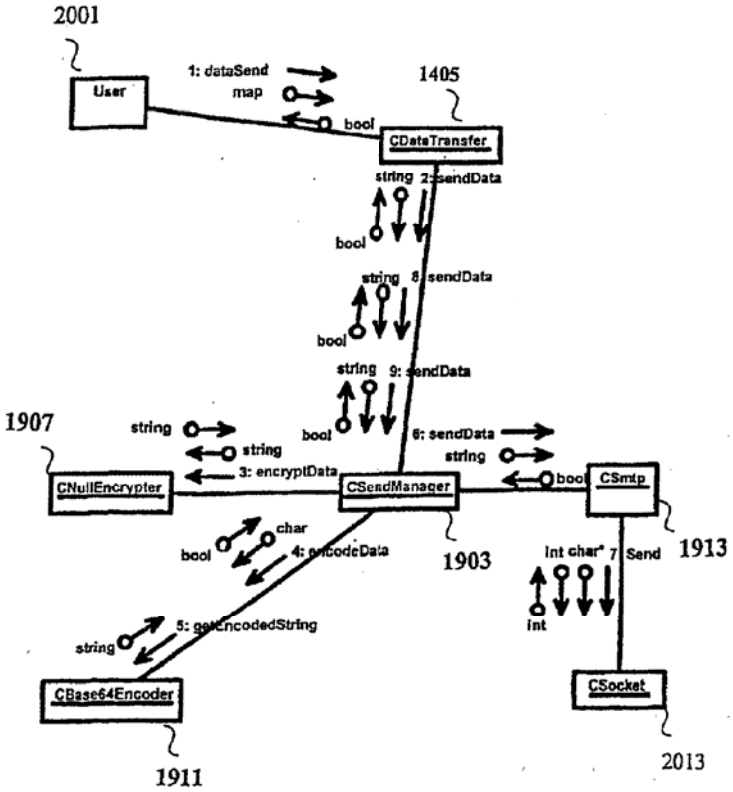


Figure 21

ForcePoint LLC
EX-1009
Page187

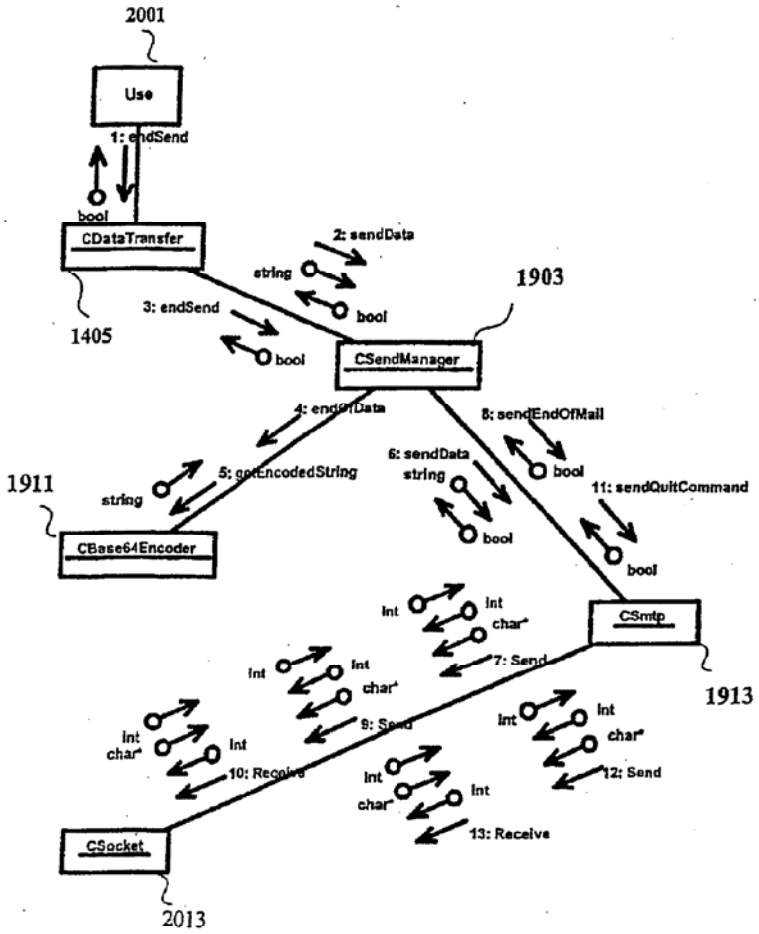


Figure 22

ForcePoint LLC
EX-1009
Page188

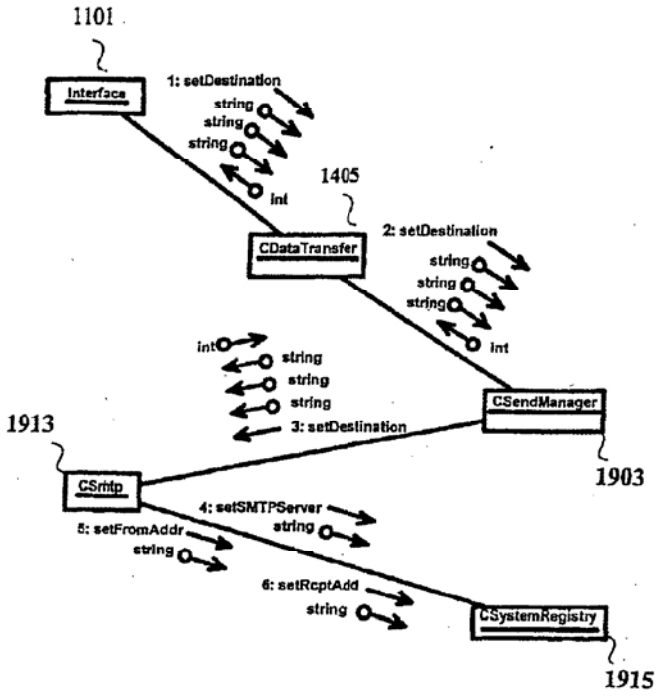


Figure 23

ForcePoint LLC
EX-1009
Page189

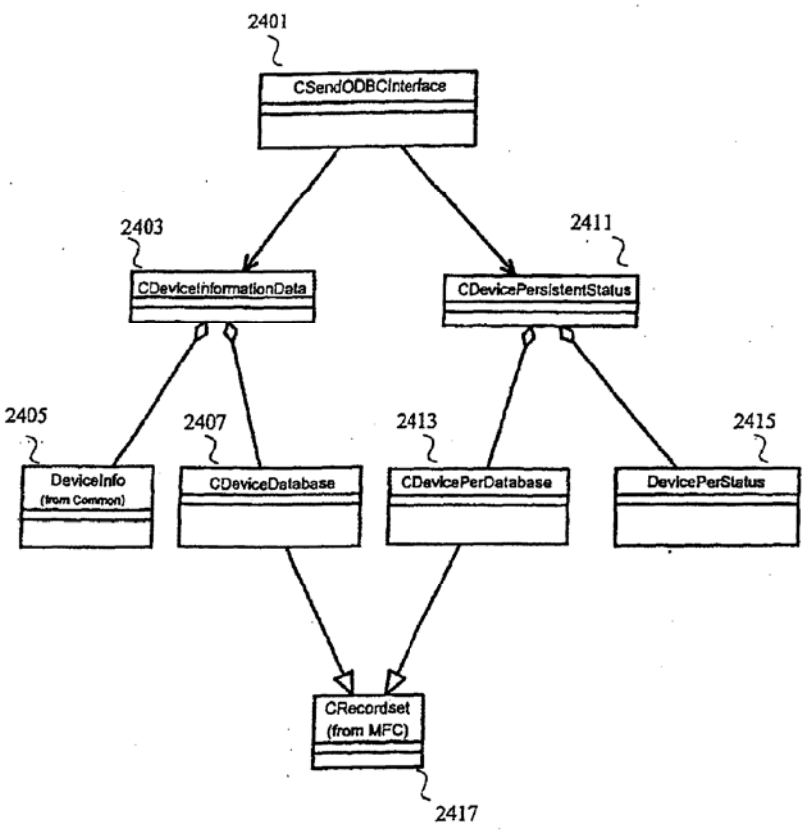


Figure 24

ForcePoint LLC
EX-1009
Page190

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 02 02 0472

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
X	EP 1 003 307 A (RICOH KK) 24 May 2000 (2000-05-24) * paragraphs [0005]-[0010] * * paragraphs [0012]-[0018] * * paragraphs [0028]-[0047] * * paragraphs [0055]-[0063] * * figures 1,5-12,16-20 *	1-15,19, 20, 24-37, 41-53	H04L12/26 H04L12/58 G05F11/34 G06F17/60
Y		16-18, 21-23, 38-40	

X	EP 1 083 486 A (RICOH KK) 14 March 2001 (2001-03-14) * paragraphs [0007]-[0010] * * paragraphs [0012]-[0018] * * paragraphs [0028]-[0045] * * paragraphs [0052],[0053] * * paragraphs [0073]-[0078] * * figures 1,5-9,12A,17 *	1-11, 13-15, 19,20, 24-27, 29-37, 41-44, 46-53	TECHNICAL FIELDS SEARCHED (Int.Cl.7) H04L G06F
Y		16-18, 21-23, 38-40	
	--- -/--		

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
MUNICH	11 December 2002	Bengi-Akyuerek, K

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

ForcePoint LLC
EX-1009
Page191

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 02 02 0472

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
A	US 5 715 393 A (NAUGLE RAYMOND H) 3 February 1998 (1998-02-03) * abstract * * claims 1,7 * * figure 2 *	6,13-18, 20-23, 31,34, 35, 37-40, 48,51,52	
A	WO 92 22033 A (BELL COMMUNICATIONS RES) 10 December 1992 (1992-12-10) * abstract * * page 10-15 * * figures 1,2 *	5,6,13, 14,30, 31,34, 35,47, 48,51,52	
A	WO 00 23894 A (PLAUM REINER ;SIEMENS AG (DE); TALANIS THOMAS (DE); BLUMENSTOCK WE) 27 April 2000 (2000-04-27) * abstract * * page 4, line 5 - page 6, line 28 * * claims 1-6 * * figures 1-3 *	15-18, 20-23, 37-40	TECHNICAL FIELDS SEARCHED (Int.Cl.7)

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
MUNICH	11 December 2002	Bengi-Akyuerek, K

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

ForcePoint LLC
EX-1009
Page192

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 02 02 0472

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

11-12-2002

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 1003307 A	24-05-2000	EP 1003307 A2	24-05-2000
		JP 2000162926 A	16-06-2000
EP 1083486 A	14-03-2001	EP 1083486 A1	14-03-2001
		EP 1246069 A2	02-10-2002
		JP 2001167020 A	22-06-2001
US 5715393 A	03-02-1998	NONE	
WO 9222033 A	10-12-1992	WO 9222033 A1	10-12-1992
WO 0023894 A	27-04-2000	DE 19848618 A1	29-06-2000
		WO 0023894 A1	27-04-2000
		EP 1131717 A1	12-09-2001
		US 2002006790 A1	17-01-2002

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

ForcePoint LLC
EX-1009
Page193

Espacenet

Bibliographic data: JP2002366525 (A) — 2002-12-20

SECURITY POLICY MAINTENANCE SYSTEM

Inventor(s): KOSHIDA TOSHIO ± (KOSHIDA TOSHIO)

Applicant(s): NEEDS CREATOR KK ± (NEEDS CREATOR KK)

Classification: **- international:** ***G06F15/00; G06F21/20; G06F21/24;*** (IPC1-7): G06F15/00

- cooperative:

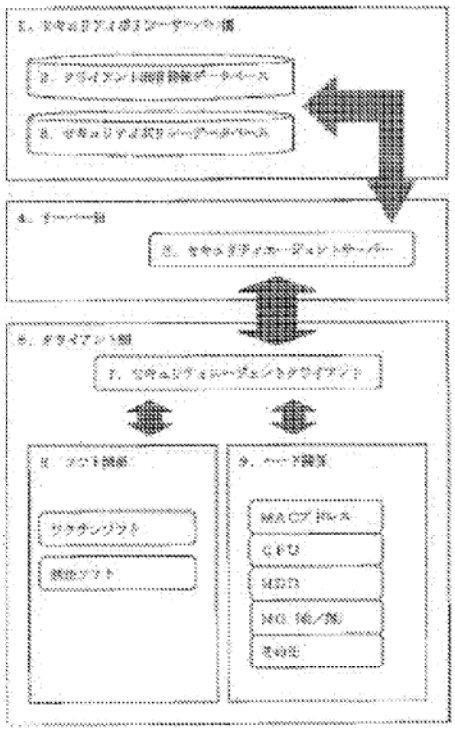
Application number: JP20010215975 20010612

Priority number(s): JP20010215975 20010612

Abstract of JP2002366525 (A)

PROBLEM TO BE SOLVED: To newly develop a system for maintaining the security policy of a network. SOLUTION: When a client 6 logs in a network, resource information of the client is acquired by a security agent client 7, transmitted to a security agent server 5, an information data base 2 intrinsic to the client is compared with a security policy data base 3 in a security policy server 1 and permission or nonpermission of network log in by the client 6 is determined. In addition, the security agent client 7 always monitors an operating state of the client 6, transmits it to the security agent server 5 and when breach of the security policy is caused, disconnects network connection by the client 6.

ForcePoint LLC
EX-1009
Page194



ForcePoint LLC
EX-1009
Page195

(19)日本国特許庁（ＪＰ）　(12)**公開特許公報**（Ａ）　(11)特許出願公開番号
特開2002－366525
（P2002－366525A）
(43)公開日　平成14年12月20日(2002.12.20)

(51)Int.Cl.[7]	識別記号	ＦＩ	テーマコード（参考）
Ｇ０６Ｆ　15/00	３３０	Ｃ０６Ｆ　15/00	３３０Ｄ　５Ｂ０８５
	３２０		３２０Ｋ

審査請求　未請求　請求項の数３　書面　（全　４　頁）

(21)出願番号　特願2001－215975(P2001－215975)

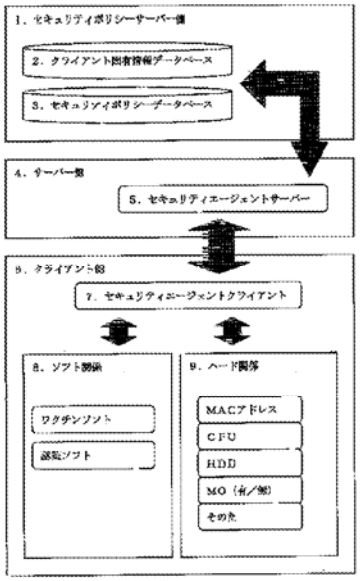
(22)出願日　平成13年６月12日(2001.6.12)

(71)出願人　501082336
ニーズクリエイター株式会社
福岡県福岡市中央区高砂１丁目21番11号
(72)発明者　越田　敏臣
福岡県福岡市中央区高砂１丁目21－11　ニーズクリエイター株式会社内

Ｆターム(参考)　5B085 AC01 AC11 AE00 BA07 BG07

(54)【発明の名称】　セキュリティポリシー維持システム

(57)【要約】
【課題】ネットワークのセキュリティポリシーを維持するシステムが無かった
【解決手段】６のクライアントがネットワークにログインする際に、７のセキュリティエージェントクライアントにて、クライアントのリソース情報を取得し、５のセキュリティエージェントサーバーに送信され、１のセキュリティポリシーサーバー内の、２のクライアント固有情報データベースと３のセキュリティポリシーデータベースと比較し、６のクライアントのネットワークログイン許可または不許可を判断する。また、７のセキュリティエージェントクライアントは、常時６のクライアントの動作状況を監視、５のセキュリティエージェントサーバーに送信し、セキュリティポリシー違反が発生した場合、６のクライアントのネットワーク接続を切断する・



ForcePoint LLC
EX-1009
Page196

【特許請求の範囲】
【請求項１】　ネットワークに参加するクライアントのセキュリティレベルにより、サーバー側よりクライアントのネットワーク参加を許可および拒否する手段と、前記のセキュリティレベル情報を取得し、クライアントとサーバー間でのセキュリティレベルの確認手段と、を備えたことを特徴とする、セキュリティポリシー維持管理装置やアプリケーションとそれを記録する媒体
【請求項２】　クライアントがネットワークに参加し、処理を継続している間も、クライアントのセキュリティレベル情報を監視し、セキュリティレベルがネットワーク参加可能レベルより低くなった場合やセキュリティレベル情報の取得が出来なくなった場合にネットワーク資源の利用を制限する手段とを備えたことを特徴とする、メール管理装置やアプリケーションとそれを記録する媒体
【請求項３】ネットワーク管理者が許可した正規クライアントと無許可のクライアントの判断手段とを備えたことを特徴とする、セキュリティポリシー維持管理装置やアプリケーションとそれを記録する媒体
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、コンピュータネットワークの利用においてネットワークに参加するクライアントセキュリティレベルにより、ネットワークセキュリティレベル管理制御装置やアプリケーションとそれを記録する媒体に関するものである。
【０００２】
【従来の技術】従来は、コンピュータネットワークを構築し運用していく場合は、事前に運用マニュアル等を作成し、その中でセキュリティ維持の為の項目を設定し、利用者のモラルの向上を行い、セキュリティレベルを維持していた。
【０００３】また、ネットワーク参加者の区別は、ネットワークログインＩＤとパスワードのみチェックで参加を許可していた為、プライベートパソコンや取り決めたセキュリティレベルを維持していないクライアントも、簡単にネットワークに参加していた。
【０００４】また、ネットワーク管理者が知らないセキュリティレベルの低いクライアントが数多くネットワークに参加する事で、ネットワークのセキュリティレベルが低下し、ウィルスへの感染や情報の漏洩などの重大な問題が発生する場合があった。
【０００５】
【発明が解決しようとする課題】本発明は、前記の課題を解決する為に、クライアントのセキュリティレベル情報取得と取得したセキュリティレベルによりネットワークへの参加を拒否および制限を行うことが出来る様にしたもので、以下の装置および方法を特徴とする。
【０００６】本発明の目的は、ネットワーク管理者が定めたセキュリティレベルの基準に至らないクライアントのネットワーク参加を制限する管理方法を提供することにある。
【０００７】
【課題を解決する為の手段】サーバーとクライアントにセキュリティチェックエージェントを常駐させ、クライアントの電源を入れ、起動した時点からセキュリティエージェントが起動する。そのセキュリティエージェントは、クライアント上にネットワーク管理者が設定した、常時正常動作が必要なアプリケーションの稼動状況とバージョン情報の取得を行う。また、その他の稼動中アプリケーション情報の取得を行い、取得した情報をログイン処理と共にサーバーに伝送する。
【０００８】サーバー側は、伝送されたクライアントセキュリティ情報および、クライアントのＭＡＣアドレスやＣＰＵおよびＨＤＤサイズ等の固有データをセキュリティポリシー設定データベースの内容と比較し、ネットワーク参加の可否を判断する。
【０００９】ネットワーク参加が許可された場合は、ネットワーク資源を利用する事が出来る様になるが、不許可の場合は、ネットワークログインを拒否する。
【００１０】また、前記のセキュリティエージェントが稼動していないクライアントからネットワークサーバーにログイン処理が行われた場合は、サーバー側に常駐しているセキュリティエージェントにより、ネットワークログイン処理を強制的に中断し、ネットワーク参加を拒否する。
【００１１】
【発明の実施の形態】ネットワーク利用者は、自己の管理している６のクライアントを起動し、起動後ＩＤとパスワードを入力して、ネットワークにログインする。
【００１２】前記のクライアントが、正規許可のクライアント場合は、７のセキュリティエージェントクライアントにより、８のソフト稼動状況や９のハードリソース情報が取得され、ログイン処理と共に７のセキュリティエージェントクライアントから５のセキュリティエージェントサーバーに送られ、６のセキュリティエージェントサーバーは、１のセキュリティポリシーサーバー上の、２のクライアント固有情報データベースと３のセキュリティポリシーデータベース情報を元に、ログインの可否を判断する。
【００１３】セキュリティレベルが基準に達していれば、５のセキュリティエージェントサーバーは、７のセキュリティエージェントクライアントに対し、ログインを許可する。
【００１４】しかし、セキュリティレベルが基準以下の場合、５のセキュリティエージェントサーバーは、７のセキュリティエージェントクライアントに対し、ログインを拒否し、ログインの拒否理由を７のセキュリティエージェントクライアントに送信する。７のセキュリティ

ForcePoint LLC
EX-1009
Page197

エージェントクライアントは、受信した拒否理由を表示しする。利用者は指示に従って対処し、再度ログイン処理を行う。

【００１５】また、個人所有の正規許可のクライアントで無い場合は、７のセキュリティエージェントクライアントが稼動していなか、または稼動していてもＭＡＣアドレスやＣＰＵおよびＨＤＤサイズとうのクライアント固有の情報が、２のクライアント固有情報データベースに登録が無い為、５のセキュリティエージェントサーバーにより、ネットワーク参加が拒否される。

【００１６】また、７のセキュリティエージェントクライアントは、ネットワークに接続中に、６のクライアントのソフトやハードのリソースに変化を検知し、５のセキュリティエージェントサーバーに通知する、５のセキュリティエージェントサーバーは、通知内容がセキュリティポリシーデータベースと比較し、セキュリティポリシーに違反があれば、６のクライアントをネットワークより強制的に切断する。

【００１７】また、５のセキュリティエージェントサーバーは、常時７のセキュリティエージェントクライアントの動作を監視し、７のセキュリティエージェントクライアントからの信号が途絶えた場合、６のクライアントのセキュリティレベルに異常が発生したと判断し、ネットワークより強制的に切断する。

【００１８】

【発明の効果】本発明のセキュリティポリシー維持システムを利用する事で、セキュリティポリシーに違反するクライアントの接続を防止し、ネットワークのセキュリティレベルを維持できる。また、各クライアントのリソース状況をデータベースに記録し管理することで、悪意を持ってＭＯやＲＡＳ環境増設を行っても、ネットワークから自動的に切り離される為、機密情報や個人情報の漏洩やウィルスの侵入を防止することができる。

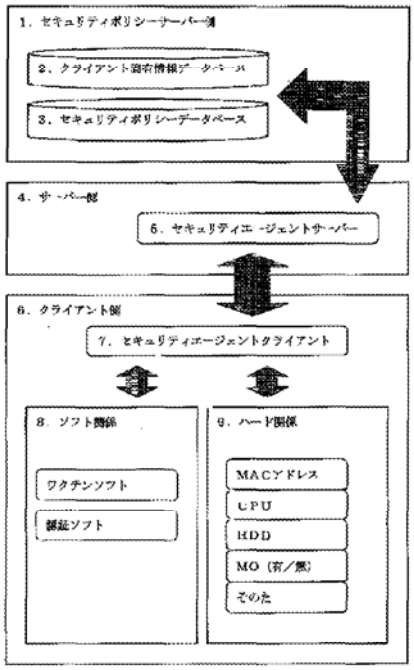
【図面の簡単な説明】

【図１】本発明の基本処理の流れを説明した図である

【符号の説明】

１　－　セキュリティポリシーサーバー
２　－　クライアント固有情報データベース
３　－　セキュリティポリシーデータベース
４　－　サーバー
５　－　セキュリティエージェントサーバー
６　－　クライアント
７　－　セキュリティエージェントクライアント
８　－　ソフト関係
９　－　ハード関係

【図１】



ForcePoint LLC
EX-1009
Page198

【手続補正書】
【提出日】平成１３年7月２３日（２００１．７．２３）
【手続補正１】
【補正対象書類名】明細書
【補正対象項目名】請求項２
【補正方法】変更
【補正内容】
【請求項２】　クライアントがネットワークに参加し、処理を継続している間も、クライアントのセキュリティレベル情報を監視し、セキュリティレベルがネットワーク参加可能レベルより低くなった場合やセキュリティレベル情報の取得が出来なくなった場合にネットワーク資源の利用を制限する手段とを備えたことを特徴とする、セキュリティポリシー維持管理装置やアプリケーションとそれを記録する媒体
【手続補正２】
【補正対象書類名】明細書
【補正対象項目名】発明の属する技術分野
【補正方法】変更
【補正内容】
【発明の属する技術分野】本発明は、コンピュータネットワークの利用においてネットワークに参加するクライアントセキュリティレベルにより、ネットワーク参加を制限する装置やアプリケーションとそれを記録する媒体に関するものである。

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
21 February 2002 (21.02.2002)

PCT

(10) International Publication Number
WO 02/14987 A2

(51) International Patent Classification[7]: G06F 1/00

(21) International Application Number: PCT/IB01/01876

(22) International Filing Date: 20 August 2001 (20.08.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/226,128 18 August 2000 (18.08.2000) US
60/259,575 4 January 2001 (04.01.2001) US

(71) Applicant *(for all designated States except US)*: **CAMELOT INFORMATION TECHNOLOGIES LTD.** [IL/IL]; Matam, Advanced Technology Center, 31905 Haifa (IL).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **GADISH, Ofer** [IL/IL]; 8, Shikma Street, 34737 Haifa (IL). **BAHARAV, Yuval** [IL/IL]; 40, Massada Street, 33076 Haifa (IL).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*
— *entirely in electronic form (except for this front page) and available upon request from the International Bureau*

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

WO 02/14987 A2

(54) Title: AN ADAPTIVE SYSTEM AND ARCHITECTURE FOR ACCESS CONTROL

(57) Abstract: A network security system and method for protecting network resources from unauthorized access and/or use. An agent device gathers data in regard to all access attempts directed at network resources and supplies this information to an analyzer device that adapts permission levels to correspond with at least the newly supplied information. The permission levels may be used by a guardian agent for the purpose of allowing or denying access to system resources. Such enforcement may further be conducted with respect to each resource access attempt in accordance with a network security policy. A control unit maintains the security policy and generates reports based on data provided from both the agent device and the analyzer device.

ForcePoint LLC
EX-1009
Page200

AN ADAPTIVE SYSTEM AND ARCHITECTURE FOR ACCESS CONTROL

CROSS REFERENCE TO RELATED APPLICATIONS

[001] This application claims priority to U.S. Provisional Application 60/226,128, filed August 18, 2000 and U.S. Provisional Application 60/259,575, filed January 04, 2001.

FIELD OF THE INVENTION

[002] The present invention relates generally to computer networks and, specifically, to access control with respect to computer networks.

BACKGROUND OF THE INVENTION

[003] Computer networks may consist of vast amounts of information and/or resources, such as files, documents, texts, databases, servers, printers, plotters, etc. (collectively "resources") shared among a large numbers of users. The resources may have a varied degree of sensitivity and may not necessarily be appropriate for use by all the users of the computer network or by users outside the network. This problem is especially pronounced in the business environment where there are global users and where business-to-business (B2B) communications are common.

[004] Thus, in order to protect resources from inappropriate use, each user in the network may be assigned access to only some, rather than all, resources on the network, effectively restricting access by each user to an appropriate subset of those resources. As a result, people with fewer access restrictions may have access to more of the network

ForcePoint LLC
EX-1009
Page201

resources, while others with more access restrictions may have access to only a limited number of resources. A member of a development group, for example, may be assigned access privileges to resources pertaining to a particular project on which he is working, while at the same time being restricted from access to other management resources. Each user is thus assigned "static" access privileges according to his perceived level or task. Access restrictions may similarly be assigned to the resources themselves. For example, a network printer may be made available to everyone on the network for printing, or it may be restricted to only those individuals who are granted special access.

[005] Typically, access permissions are controlled at the resource level. Thus, each resource would have a corresponding access control list (ACL) generated either during the creation of the resource or at a later date. An ACL usually comprises a list of access entries, each such access entry containing a user's name and his/her associated permissions/restrictions. In some instances, the access entry may comprise a user group (such as accounting, engineering, marketing, etc.) and the associated access permissions/restrictions for that group. The permissions/restrictions typically allow/prevent access to the resource, or allow/deny the performance of various operations by, or on, the resource, such as deleting, reading, writing, or otherwise using the resource.

[006] In certain operating systems (OSs), such as the UNIX® operating system, the ACL has been simplified to allow only three predefined accessibility levels of users: the owner of the resource, the owner's group, and the world, which would include anyone requesting access. For each of these user levels, three basic access permissions may generally be possible: "read", "write", and "execute".

ForcePoint LLC
EX-1009
Page202

[007] One known method for carrying out access control on a network may be as follows: A person, X, is accepted into an enterprise or organization network, at which point he is associated with a user name and possibly added to one or more user groups. When person X requests access to a certain resource, the access list associated with that resource is consulted and searched for either the user name or the user group(s) associated with person X. Depending on the relevant ACL, access to the particular resource is either permitted or denied. In fact, in conventional systems, the only possible conclusion for a given access request is either to permit or to deny access.

[008] In such a system, the resultant security policy, which may be seen as a collection of all potential access permissions and denials, is distributed throughout the system, with little or no capability for effective management. Some systems include "agents" that monitor access and consult a more global policy, normally centrally located, to each permission grant or denial.

[009] However, known network security methods present several problems. First, since ACLs are typically input manually, the creation of the lists may be time consuming. As the number of users and resources grows, the task becomes more cumbersome and more prone to mistakes such as inappropriate exclusions, accidental inclusions, etc. Moreover, the person who typically sets up the listing, e.g., the system administrator, may not be aware of every user and/or resource on the network. Furthermore, even if the system administrator is aware of the relevant user and resource, he may not know the job requirements and applicability of each, and so he may not be able to determine the appropriate ACL.

[010] Yet another problem with conventional network security systems is the inflexibility of the ACL, which, by the nature of the input process, is predefined. Each

ForcePoint LLC
EX-1009
Page203

time a change occurs in the system, whether in regard to a user, a resource, or permission therebetween, the ACL must be amended. Furthermore, since the listing is predefined, exceptions are difficult to implement. For example, if a user changes jobs, his group must be manually changed for him to gain access to the shared resources of his new group. If the user is performing a job temporarily, permissions may be needed for only a limited period of time. Certain permissions for him would have to be granted to perform the job and then revoked when no longer needed, requiring that someone remember to revoke the permissions at the later date.

[011] Thus, there exists a need for a more efficient, better defined access control method and for an associated system architecture that provides for flexible, adaptive use.

SUMMARY OF THE INVENTION

[012] An object of the present invention is to provide an access control system and architecture for accessing resources. It is another object of the invention to provide an access control system and architecture for accessing resources such as databases, files, computer peripherals and others.

[013] An embodiment of the present invention, therefore, provides a system adapted for controlling access by one or more users to one or more resources. The system includes at least one agent, which collects data about access attempts concerning the resources, and at least one access analyzer, which receives and processes the collected data. The access analyzer analyzes at least the collected data and generates permission levels based on the analysis.

[014] In accordance with the invention, an "access attempt" is an attempt to gain access to any of the resources on the system, regardless of whether the access is ultimately permitted or not. Thus, an access attempt includes the situations where access

ForcePoint LLC
EX-1009
Page204

is granted, and also where access is denied. The data collected by the agent can be, for example, behavioral data concerning the users as well as data concerning the resource itself. More specifically, examples of the collected data include; the access distribution of the resource(s), that is, how each resource is allocated to the user(s); the level and frequency of access attempts initiated by particular user(s), etc. The collected data can also include information about activities such as how much CPU time each user utilizes and/or data regarding I/O and application usage.

[015] The permission levels may be access control permission levels, wherein the permission levels are presented as numbers within a given range, and wherein the likelihood that the access attempt is to be permitted is determined based on the value of the permission level. For example, the permission levels may be normalized to a range between 0 and 1 and the access can be permitted/denied depending on whether the permission level is above/below a certain threshold.

[016] The agent may include at least an enforcement means adapted to control access to the resources based on at least the permission levels. A system in accordance with the present invention may further include one or more controllers adapted to provide one or more rules to the agent, wherein the rules may be access control rules.

[017] Additionally, the enforcement means may be adapted to control access to the resources based on at least the permission levels and the rules and, the enforcement means need not necessarily be located within the agent. The enforcement means can be located external to the immediate network and its servers and enforce access to the resources through remote means. In further embodiments of the invention, the system may also include a discovery unit adapted to provide information to the controller,

ForcePoint LLC
EX-1009
Page205

wherein the information may be data concerning the users and the resources. The discovery unit may include means for automatically gathering the information.

BRIEF DESCRIPTION OF THE DRAWINGS

[018] The object and features of the present invention will become more readily apparent from the following detailed description of the preferred embodiments taken in conjunction with the accompanying drawings in which:

[019] **FIG. 1** is a block diagram illustration of an access architecture, constructed and operative according to an embodiment of the present invention;

[020] **FIG. 2** is a block diagram illustration of an access control system, implemented using the access architecture of FIG. 1, constructed and operative according to an embodiment of the present invention; and

[021] **FIGS. 3A and 3B** are block diagram illustrations of alternative access control systems, implemented using the access architecture of FIG. 1, constructed and operative according to an embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[022] A preferred embodiment of the present invention is discussed in detail below. While specific configurations are discussed, it should be understood that this is done for illustration purposes only. A person skilled in the relevant art will recognize that other components and configurations may be used without departing from the spirit and scope of the invention.

[023] Reference is now made to FIG. 1, which is a block diagram illustrating the operation of the invention and an example of access architecture 10. Architecture 10 illustrates one embodiment of the present invention; other embodiments are also described below.

ForcePoint LLC
EX-1009
Page206

[024] In order to facilitate the discussion of a preferred embodiment, the following is a list of relevant terms and associated definitions: "Resources" may be services, documents, data, files, databases, or any portion of such services, documents, data, files, databases, as well as peripheral devices or any other type of computer resources. "Access Attempt", as mentioned above, may be an attempt to use or otherwise gain access to a particular resource. "User(s)" may be any person, user group, or other resource wishing to access a resource; persons such as members of a company with which the resource resides, associates of the company (e.g., in a B2B situation), Internet browsers, etc. or programs or other resources. "User groups" may be teams of users with some actual or perceived shared trait, such as shared responsibilities, e.g., an accounting group, shared location, e.g., Haifa group, shared hierarchy in the corporation, e.g., a management group, etc. "Policy" may be a list of rules and "security policy" may be a list of rules that control the access privileges to resources. "Rules" may comprise at least user or user group names and/or associated access permission, denial or other result. "Administrators" may be persons who administer or manage computer systems such as those described herein. A "security policy table" may be a table that lists access options and other rules related to resource security. "Permission levels" refer to the likelihood that accesses to the resource will be required and/or granted. The permission levels may be likelihood estimates, possibly discretized and normalized to a specific range, for example, between 0 and 1. Such a permission level may be specific to the user, his group, time-of-day, location, or other parameters. An "analyzing algorithm" may be an algorithm that, based on gathered data, for example access requests, learns about activities and at least therefrom creates permission levels. "Adaptive permission levels" or "APL" may be permission levels generated periodically by an analyzing algorithm.

ForcePoint LLC
EX-1009
Page207

An "agent" may be a software utility that collects data about the activities of the computer; a "guardian agent" is a type of agent that is capable of enforcement based on at least security policy rules. A guardian agent may further be capable of enforcement based, in addition, on permission levels.

[025] Illustrated in FIG. 1 is an exemplary system in accordance with the present invention. Three elements comprise an architecture 10: an Agent 100, a Control Unit 110, and an Access Analyzer 120. Agent 100 monitors access attempts 108 directed to particular resources, (not shown), and provides a periodic event audit trail 102 to access analyzer 120, reports alarms 104 to control unit 110 and, if it is a "Guardian Agent", enforces access restrictions 106 to the resource. Event Audit Trail 102, includes information regarding the access attempt(s), including but not limited to such things as whether the access was permitted, time of the access attempt, etc. Access analyzer 120 analyzes event audit trail 102, possibly using a first security policy 122 provided by control unit 110, and responds periodically by sending a list of permission levels 134 to agent 100. Access analyzer 120 also sends statistical information 132 to control unit 110. Based on permission levels 134, received from access analyzer 120 and optionally second security policy 122' received from control unit 110, agent 100, if it is a guardian agent, can provide enforcement 106. Access enforcement 106 includes enforcing the security control of the resources by permitting, alerting, denying, or otherwise controlling access to the resource.

[026] First and second security policy arrows 122 and 122' can represent the same security policy information from control unit 110 to access analyzer 120 and agent 100, respectively, or either the first or second security policy, 122 or 122', can comprise a subset of the other. Independently sending/receiving a security policy allows for the use

ForcePoint LLC
EX-1009
Page208

of multiple agents 100, access analyzers 120, and control units 110 which all either send or receive united security policy 122, or 122'. Control unit 110 may use the alarms 104 received from agent 100, the statistical information 132 received from Access analyzer 120, and other user inputs (not shown) to create reports 126. Reports 126 provide all, or some, of the input information in a user defined format.

[027] In another embodiment of the present invention (not shown), first security policy 122 can be provided to agent 100 by way of access analyzer 120. In other words, access analyzer 120 receives first security policy 122 from control unit 110 and then sends it along to agent 100. According to this embodiment, it would not be necessary to provide second security policy 122' from Control Unit 110 to Agent 100.

[028] According to the architecture 10, described above, a system in accordance with the present invention may be used for various operations, including analysis, intrusion detection, website user profiling, database access, or any other resource access application, such as access control.

[029] FIG. 2, to which reference is now made, provides a relatively detailed illustration of an exemplary configuration of architecture 10, herein referred to as an access control architecture, or access system 20, which is constructed and operative in accordance with an embodiment of the present invention. Access system 20 comprises one or more agents 100, one or more access analyzers 120, one or more central control servers 110, an optional security policy (not shown) originating in control server 110, and optionally one or more auto discovery units 22. Access system 20 resides, for example, in a university, business, or any other organization, that has one or more servers 18, 18' and 18" which may communicate with other devices through either non-secure or secure channels (e.g., encrypted channels). Servers 18, 18' and 18'' include any computer

ForcePoint LLC
EX-1009
Page209

resources that serve as gateways to other computer resources or services. Exemplary servers shown in FIG. 2 are databases 18', file servers 18", and personal computers 18, but other, similar machines, such as printer servers (not shown), could also be connected to system 20 in accordance with the present invention.

[030] In accordance with the present invention, system 20 can utilize one or more adaptive algorithms. For example, system 20 may be dedicated to the adaptive access control of resources on a network, as depicted in architecture 10 (FIG. 1). Optionally, for secure embodiments of the present invention, system 20 may also comprise a key authority (not shown), which can reside on control server 110 or on another appropriate platform. In a secure embodiment of the present invention, all communications to/from each device can be encrypted in accordance with conventional encryption tehniques. For example, each of the devices shown in FIG. 1, for example, Agent 100, Access Analyzer 120 and Control Unit 110, can be equipped with both encryption and decryption tools in order to encrypt the information it sends, and decrypt the information it receives.

Agent

[031] Agents 100, which may also be Guardian Agents as described above, reside on servers 18, 18' or 18" and monitor access attempts that occur on servers 18, 18' or 18" and gather data related to those access attempts, data such as the name of the user attempting access, the machine from which access is attempted, the time of day of the access attempt, the resource to which access is attempted, the type of access attempted, etc. The gathered data may then be sent as event audit trail 102 (FIG. 1) to access analyzer 120 to be used as input for an analyzing algorithm run by access analyzer 120. The analyzing algorithm can use at least the gathered data to generate permission levels 134 (FIG. 1).

ForcePoint LLC
EX-1009
Page210

[032] In further embodiments, based on at least the permission levels 134, guardian agents 100 can protect the resources stored on servers 18, 18' or 18" by enforcing the permission levels 134. Enforcement may be performed by guardian agent 100 or by an enforcement unit (not shown) provided either within guardian agent 100 or as a separate unit.

[033] In yet another embodiment of this invention, event audit trail 102 is used by access analyzer 120 to generate the underlying system permissions, e.g., ACLs, that are used to update the system ACLs. In this embodiment, agent 100 may not necessarily be a guardian agent. In another embodiment of this invention, the consulting facilities of a computer system may be used. In such a case as a response to an access attempt 108 the system will respond with a query to the consulting facilities. The consulting facilities, based on permission levels provided by access analyzer 120 may respond with the required response relative to access attempt 108. In yet another embodiment of this invention, agent 100 may be implemented as a "proxy", i.e., a unit through which all the network traffic flows and which may not necessarily be part of the system it is protecting, and access attempts 108 will be handled from the "proxy" unit.

[034] Also, agents 100 can reside on control server 110 or any other appropriate platform. That is, it is not necessary for Agents 100 and Control Units 110 to be separate machines, their functionalities can be combined into a single machine.

[035] In other embodiments, based on at least the permission levels 134 and the security policy 122, 122', guardian agents 100 may protect the resources stored on servers 18, 18' or 18" by executing and enforcing access permissions and restrictions, and possibly notifications to other system resources, based on both the permission levels 134 and security policies 122, 122' to control access to the resources. As mentioned

ForcePoint LLC
EX-1009
Page211

above, enforcement may be executed by guardian agent 100 or by an enforcement unit (not shown) within guardian agent 100, or as a separate unit.

Access Analyzer

[036] Access Analyzer 120 may be in communication with one or more agents 100 and each access analyzer 120 can support and control each of these agents 100, depending on the load of each agent 100 and the strength of access analyzer 120. Access analyzer 120 may further receive event audit trail 102 by a push, i.e., where the sender initiates the transfer, or pull, where the receiver initiates the transfer, method to transfer the event audit trail data from each agent 100, applying the data to an analyzing algorithm. For example, the transfer of event audit trail 102 may take place on a cyclic basis (e.g., once a day) and the analyzing algorithm may then be executed after the new event audit trail 102 data has been received.

[037] Access analyzer 120 can utilize any learning algorithm adapted for access control, an example of which is described in the co-pending U.S. Patent Application, filed on the same date herewith, entitled "Permission Level Generation Based on Adaptive Learning", and which is assigned to the same common assignee as the present application, and is incorporated herein by reference in its entirety for all it discloses.

[038] Access analyzer 120, based on security policy 122 and event audit trail 102, is operable to estimate the probability of an access attempt to a system resource occurring and, subsequently, define the most up-to-date permission levels 134. Permission levels 134, which are an output of the analyzing algorithm utilized within Access Analyzer 120, may be transmitted to agent 100, which may receive and/or transmit data to one or more access analyzers 120.

ForcePoint LLC
EX-1009
Page212

Central Control Unit

[039] Control unit 110 may comprise means for interacting with access analyzers 120 and agents 100, and can manage activities within the system architecture 10 (FIG. 1) and system 20 (FIG. 2) having a single security policy, i.e., a combination of security policies 122, 122'. In other embodiments of the present invention, one or more control servers 110 may handle a single security policy. Control server 110 may control the system configuration, security policy, and response to reported events. When access attempts 108 occur, they may be reported to control unit 110, and, based on the nature of the attempt, control unit 110 may notify the appropriate person(s) or program(s) by e-mail or other form of communication. Control unit 110 may include a database, a report generation engine, and a scheduler.

[040] In a further embodiment in accordance with system 20, agents 100 may monitor access attempts 108 that occur on servers 18, 18' and 18" and gather data related to those attempts. The access attemtps and data that are monitored may relate to user or resource activities, or to any other operations that may occur on servers 18, 18' or 18". The access attempts 108 that are monitored may include resource retrieval and/or usage of, for example, documents, files, databases, computer peripherals, etc., resource accesses, logins, internal communication problems, access times and types, etc. Event audit trail 102 may include the number of times a resource is accessed, the users that access a specific resource, access time, type of access, any type of statistical data related to the access attempt, etc.

[041] Event audit trail 102 may be used as an input for the analyzing algorithm run by access analyzer 120 as mentioned above. In some embodiments of the invention, the analyzing algorithm uses at least event audit trail 102 to generate permission levels 134

ForcePoint LLC
EX-1009
Page213

for respective accesses to resources. Permission levels 134 may be specific to each type of access, they may be time dependent, and/or they may correspond to each user and each resource.

[042] Agent 100 may receive permission levels 134 from access analyzer 120 and may receive security policy 122', in the form of a table, or some other format, from control unit 110. In some embodiments of the invention, rules related to the security policy may be defined in control unit 110 and enforced by a guardian agent type of agent 100.

[043] Based on at least permission levels 134 and second security policy 122' guardian agent 100 may execute and enforce the overall access requirements of the resource, thereby protecting the resources stored on servers 18, 18' or 18" from unauthorized access or use. Security policy rules may include a first threshold below or above which an alarm will be generated to notify the control unit 110 of an access attempt to a resource. Security policy rules may also include a second threshold below or above which access attempts to a resource will be denied. An example of a security policy including the security policy rules just described is in the co-pending U.S. Patent Application, filed on the same date herewith, entitled "A Method and Apparatus for a Security Policy", and assigned to common assignee of the present invention, and which is incorporated in its entirety herein by reference for all that is disclosed.

[044] Enforcement may comprise two different operations. For example, enforcement can include; 1) allowing operations that are permitted by both the security policy and the permission levels, if both exist, or; 2) blocking operations that are not permitted or are considered suspicious beyond a second threshold level mentioned above. The second operation may further include the generation of a different alarm if an operation is considered suspicious as is beyond a first level threshold.

ForcePoint LLC
EX-1009
Page214

[045] Enforcement may be performed by guardian agent 100 or by an enforcement unit (not shown) within guardian agent 100. Alternatively, agents 100 may reside on control unit 110 or any other appropriate platform having access to system resources.

[046] The security policy rules and permission levels 134 may correspond to a user, a group of users, a resource, a group of resources or a combination of user(s) and resources(s). The security policy rules may also be applied on an access type and/or time basis, and/or may be applied on the basis of access parameter availability, such as location. User groups may be created by applying an algorithm, observing formal or informal hierarchy, or other method known in the art.

[047] Security policy rules may correspond to a resource, a user, or a <resource, user> pair, possibly in combination with a particular time, and access type as mentioned above. Agent 100 may determine which rules apply to each resource, each user, and each <resource, user> pair at each time. Enforcement determinations can be made on the basis of at least the rules and/or on other factors, such as location. Conflicts between rules that are defined for the same <resource, user> pair may be resolved in the security policy rules or flagged by the system, which can determine how to handle conflicts. The system may, for example, always follow the first security policy rule matching the access attempt. In another embodiment, the system may, for example, always follow the stricter of conflicting security policy rules.

[048] In further embodiments of the invention, agent 100 may protect specific resources by applying adaptive access control only to specific resources of the existing security system. In such embodiments, agent 100 does not replace or use any of the existing security subsystem. Instead, agent 100 may, in addition to following the specific resource rules, continue to enforce the system's existing rules and thus, may not

ForcePoint LLC
EX-1009
Page215

permit anything that is blocked by the existing security subsystem. For example, if the operating system (OS) already permits/denies access to certain resources based on its own independent rules, the adaptive access control system of the present invention will not override the OS's rules and allow access to users that would otherwise not be permitted access.

[049] By being an extension of the existing system, agent 100 can further provide a uniform interface between the various systems since the rules of other existing systems are incorporated without the need to interface with those systems directly. Interfacing to other systems in this manner satisfies the definition of unified and universal security policy rules.

[050] There may be an agent 100 for each platform or application, and agent 100 may be implemented as an extension of the operating system (OS), a database, a Web-server, or an application.

[051] Discovery units 22 may comprise a tool used by control unit 110 to obtain information concerning users and resources on servers 18, 18' or 18". Discovery units 22 can receive instructions from control unit 110, when appropriate. For example, discovery unit 22 can collect information regarding which users are defined on servers 18, 18' or 18" and which resources are defined, or any other information that may be useful to control unit 110. As a further example, in some embodiments of the invention, discovery units 22 can report which users are logged-on the respective system. Discovery units 22 can gather information automatically or in response to a request from control unit 110 and Discovery units 22 may be a part of agent 100 or they can be a stand-alone units.

ForcePoint LLC
EX-1009
Page216

[052] Access analyzer 120 can be in communication with agent 100 and receive event audit trail 102. Event audit trail 102 can be used as an input to access analyzer 120 and used in accordance with an adaptive access control analyzing algorithm, in which at least some of the output of the analyzing algorithm, for example permission levels 134, may be used as part of the enforcement 106.

[053] In order for access analyzer 120 to generate access control permission levels 134, the analyzing algorithm can use "Knowledge" pertaining to user activity. Knowledge is derived from the data gathered by agents 100. The data used in gaining "Knowledge" about user activity can be transferred from agents 100 to access analyzers 120 as frequently, or infrequently, as necessary, such as daily, bi-weekly, etc. Updated data can be matched with past (known) user behavior patterns, and the behavior patterns can be updated by access analyzer 120. By consulting the user behavior patterns, it is possible to analyze and determine what each user does, what the relationships between users are, and which resources are likely to be used by various users (or user groups) in the future. As mentioned above, an exemplary analyzing algorithm is described in the co-pending U.S. Patent Application, filed on the same date herewith, entitled "Permission Level Generation Based on Adaptive Learning".

[054] Furthermore, access analyzer 120 can generate permission levels, possibly by calculating the likelihood that a particular access attempt should be permissible. The permission levels are numbers in a given range, wherein the higher the number the more likely the access attempt is to be permitted. The permission levels may be normalized to a range between 0 and 1, or any other appropriate range or scale.

ForcePoint LLC
EX-1009
Page217

[055] In some embodiments of the invention, access control determinations can be made in run-time by agent 100. In such an instance, access analyzer 120 provides the permission levels 134 to agent 100 in advance or in generally real-time.

[056] Providing organizational structure of users information of the system to the analyzing algorithm is optional. In other words, the analyzing algorithm can operate using only the data gathered by agent 100. Nevertheless, the analyzing algorithm can accept additional data, such as organizational structure information, if provided. Additional data, such as the OS permissions, or feedback from users, may improve the quality of the results, shorten the run time of the analyzing algorithm, or reduce the numbers of runs of the analyzing algorithm until convergence, or a good result, is achieved. Conversely, false input can reduce quality or increase run-time. However, given that the algorithm can resolve inaccuracies by "learning", the false input may not cause completely incorrect determinations, and effects of such will be practically eliminated over time.

[057] Control unit 110 can request that agent 100 gather information upon demand. Control unit 110 can also act as a cryptographic key manager, serving as a certificate authority, and can, further maintain a list of system administrators and their associated privileges.

[058] When access attempts 108 occur, they may be reported to control unit 110, and, based on the nature of the event, control unit 110 may notify the appropriate person(s) and/or log the events outlined in the reports for future reference. Control unit 110 can be operable to prepare and generate reports, schedule activities for execution, save the reports in an archive, and/or optionally distribute the reports by email, links, or other means of communication, to a recipient list, and/or link to them.

ForcePoint LLC
EX-1009
Page218

[059] In accordance with an embodiment of the present invention, the following is an exemplary access control scenario: Agent 100 monitors a request for access to a certain resource. Agent 100 either has or receives security policy 122' (including rules) from control unit 110. Agent 100 also either has or receives permission levels 134 from access analyzer 120. Based on permission levels 134 and security policy 122', agent 100, if it is a guardian agent, provides enforcement 106. Enforcement 106 comprise of blocking access, permitting access, reporting the access attempt, etc.

[060] Access analyzer 120 receives from agent 100 data about the access attempt and executes the analyzing algorithm. The output of the analyzing algorithm may be permission levels 134, which are transferred periodically to agent 100. The activities of agent 100 and access analyzer 120 may be time-independent with respect to each other. For example, while agent 100 operates on each and every access attempt 108, access analyzer 120 may operate periodically, based on sufficient information collected from event audit trail 102, or at the conclusion of a predefined period of time.

[061] Control unit 110 can be notified of the access attempt and any subsequent confirmation of enforcement and can generate a report based on the data for on-line review or for distribution.

[062] An important feature of the present invention is that each of the parts -- agent 100, access analyzer 120, and control unit 110 -- can operate independently. That is, agent 100 can continue operating even if its connection to access analyzer 120 is not operative. For example, Agent 100 can continue gathering event audit trail data and use its latest stored version of permission levels 134 even if no updated permission levels are being provided. Access analyzer 120 can perform the analyzing algorithm using the latest event audit trail data it has received, whether or not agent 100 is currently

ForcePoint LLC
EX-1009
Page219

communicating with access analyzer 120. Likewise, access analyzer 120 can perform the analyzing algorithm using the latest security policy 122 it has received, whether or not control unit 110 is currently communicating. Control unit 110 can be used at any time to set overall security policy 122, 122', which is transmitted to access analyzer 120 and agent 100, when possible. Finally, statistical information 132 and alarms 104 can be received by control unit 110 when any disrupted connections are eventually restored, and reports 126 can be generated independently, periodically or upon request.

[063] System 20 is further operable on alternative architectures, such as those illustrated in FIGS. 3A and 3B, which show architectures 30 and 40, respectively, to which reference is now made. Architecture 30 is a simple configuration that may be applicable for small businesses. As shown in FIG. 3A, architecture 30 can comprise control unit 110 and file server 18, with the components of agent 100 and access analyzer 120 both residing on server 18. Architecture 30 can provide the compactness and flexibility needed for small computing environments.

[064] Alternatively, architecture 40, shown in FIG. 3B may be applicable for application service providers (ASPs), and may operate on a local or non-local system, such as the Internet or web, and may operate on a direct line, without intermediate providers. Architecture 40 may comprise at least one file server 18, one or more optional firewalls 28, and an application server 32. As noted in FIG. 3B, in order to adapt to the appropriate architecture and/or system, the components of agent 100 may be operable from file server 18. Access analyzer 120 and control unit 110 may reside on application server 32. Application server 32 may be connected to file server 18 through firewalls 28 and over the Internet. This will allow for a remote implementation of the security system by an application provider.

ForcePoint LLC
EX-1009
Page220

[065] It will be appreciated by persons skilled in the art that the present invention is not limited by what has been particularly shown and described herein above. Rather the scope of the invention is defined by the claims that follow.

ForcePoint LLC
EX-1009
Page221

WHAT IS CLAIMED IS:

1. A computer network with a security system, the security system comprising:

at least one agent operable to identify an access attempt directed to a resource on the network and further operable to respond to said access attempt in accordance with at least permission levels; and

at least one access analyzer operable to analyze an event audit trail and provide said permission levels to said agent.

2. The computer network of claim 1, further comprising at least one control unit operably connected to said agent and said access analyzer.

3. The computer network of claim 2, wherein said control unit comprises at least a security policy.

4. The computer network of claim 3, wherein said guardian agent implements a unified security policy which is a unification of the said systems original security policy and an security policy operable with at least said permission levels.

5. The computer network of claim 4, wherein said unified security policy implements the most restrictive rules.

ForcePoint LLC
EX-1009
Page222

6. The computer network of claim 4, wherein said unified security policy implements the most permissive rules.

7. The computer network of claim 3, wherein said security policy comprises at least one security rule.

8. The computer network of claim 7, wherein said security rule defines at least one response to said access attempt.

9. The computer network of claim 8, wherein said response comprises at least allowing an access.

10. The computer network of claim 8, wherein said response comprises at least denying an access.

11. The computer network of claim 8, wherein said response comprises at least notification of an access.

12. The computer network of claim 8, wherein said response is adaptive in accordance with at least said permission levels.

ForcePoint LLC
EX-1009
Page223

13. The computer network of claim 12, wherein said adaptive response is further determined in accordance with at least one activation threshold.

14. The computer network of claim 13, wherein said response comprises at least allowing an access, determined at least in accordance with a first activation threshold.

15. The computer network of claim 13, wherein said response comprises at least denying an access, determined at least in accordance with a first activation threshold.

16. The computer network of claim 13, wherein said response comprises at least notification of an access, determined at least in accordance with a second activation threshold.

17. The computer network of claim 2, further comprising a discovery unit operable to provide information to said control unit.

18. The computer network of claim 17, wherein said information includes data relative to at least a user of said network.

19. The computer network of claim 17, wherein said information includes data relative to at least a resource accessible through said network.

ForcePoint LLC
EX-1009
Page224

20. The computer network of claim 17, wherein gathering of said information is automatic.

21. The computer network of claim 2, wherein said access analyzer provides said agent with only deny or permit status for each access attempt.

22. The computer network of claim 1, further comprising at least one control unit operably connected to said access analyzer.

23. The computer network of claim 22, wherein said control unit comprises at least a security policy.

24. The computer network of claim 23, wherein said guardian agent implements a unified security policy which is a unification of the said systems original security policy and an security policy operable with at least said permission levels.

25. The computer network of claim 24, wherein said unified security policy implements the most restrictive rules.

ForcePoint LLC
EX-1009
Page225

26. The computer network of claim 24, wherein said unified security policy implements the most permissive rules.

27. The computer network of claim 23, wherein said security policy comprises at least one security rule.

28. The computer network of claim 27, wherein said security rule defines at least one response to said access attempt.

29. The computer network of claim 28, wherein said response comprises at least allowing an access.

30. The computer network of claim 28, wherein said response comprises at least denying an access.

31. The computer network of claim 28, wherein said response comprises at least notification of an access.

32. The computer network of claim 28, wherein said response is adaptive in accordance with at least said permission levels.

ForcePoint LLC
EX-1009
Page226

33. The computer network of claim 32, wherein said adaptive response is further determined in accordance with at least one activation threshold.

34. The computer network of claim 33, wherein said response comprises at least allowing an access, determined at least in accordance with a first activation threshold.

35. The computer network of claim 33, wherein said response comprises at least denying an access, determined at least in accordance with a first activation threshold.

36. The computer network of claim 33, wherein said response comprises at least notification of an access, determined at least in accordance with a second activation threshold.

37. The computer network of claim 22, further comprising a discovery unit operable to provide information to said control unit.

38. The computer network of claim 37, wherein said information includes data relative to at least a user of said network.

39. The computer network of claim 37, wherein said information includes data relative to at least a resource accessible through said network.

ForcePoint LLC
EX-1009
Page227

40. The computer network of claim 37, wherein gathering of said information is automatic.

41. The computer network of claim 22, wherein said access analyzer provides said agent with only deny or permit status for each access attempt.

42. The computer network of claim 1, wherein said agent responds to said access attempt in accordance with a result of comparing at least one permission level to at least one threshold.

43. The computer network of claim 42, wherein said agent permits access to said resource if the permission level exceeds a first threshold.

44. The computer network of claim 42, wherein said agent denies access to said resource if the permission level does not exceed a first threshold.

45. The computer network of claim 42, wherein said agent sends a notification of said access attempt to said control unit if the permission level is below a second threshold.

ForcePoint LLC
EX-1009
Page228

46. The computer network of claim 1, wherein said access attempt comprises at least one of: a user, a resource, a location, an access type, a time.

47. The computer network of claim 46, wherein said user may be one or more users.

48. The computer network of claim 46, wherein said resource may be one or more resources.

49. The computer network of claim 46, wherein said location may be one or more locations.

50. The computer network of claim 46, wherein said access type comprises at least one of the following; read, write, modify, execute, delete, rename, take ownership of, change permissions of, or create said resource.

51. The computer network of claim 1, wherein said access attempt comprises at least a time scope.

52. The computer network of claim 1, wherein said event audit trail is derived from at least one access attempt.

ForcePoint LLC
EX-1009
Page229

53. The computer network of claim 52, wherein said event audit trail is provided to said access analyzer on a periodic basis.

54. The computer network of claim 52, wherein said event audit trail is periodically solicited by said access analyzer.

55. The computer network of claim 52, wherein said access attempt is ignored by said agent with respect to at least one access attempt.

56. The computer network of claim 1, wherein said permission levels are derived from at least likelihood estimates that a future access attempt should be permitted.

57. The computer network of claim 1, wherein said agent comprises at least one enforcement means for controlling access to said resources based on at least said permission levels.

58. The computer network of claim 1, wherein said permission levels are generated periodically based on a currently available event audit trail.

ForcePoint LLC
EX-1009
Page230

59. The computer network of claim 1, wherein said agent and said access analyzer reside on a single computer.

60. The computer network of claim 2, wherein said agent and said control unit reside on a single computer.

61. The computer network of claim 2, wherein said access analyzer and said control unit reside on a single computer.

62. The computer network of claim 2, wherein said agent, said access analyzer and said control unit reside on a single computer.

63. The computer network of claim 1, wherein said permission levels generated by said access analyzer are used for the purpose of updating at least an access control list of at least a system resource.

64. The computer network of claim 1, wherein said access analyzer provides permission level information to at least a system consulting unit.

65. The computer network of claim 64, wherein in response to said access attempt a query is sent to said system consulting unit.

ForcePoint LLC
EX-1009
Page231

66. The computer network of claim 65, wherein in response to said query asid system consulting unit provides a response corresponding at least to said permission levels.

67. The computer network of claim 1, wherein said agent is implemented as a proxy server.

68. A system comprising a computer network with a distributed security system, the system further comprising:

at least one agent connected to the Internet, said agent being operable to identify an access attempt directed to a resource on the computer network and further operable to respond to said access attempt in accordance with at least permission levels;

at least one access analyzer connected to the Internet, said access analyzer being operable to analyze an event audit trail and provide said permission levels to said agent; and

a communication device operable to provide communication between said agent and said access analyzer.

69. The system of claim 68, wherein the agent is connected to the Internet through a firewall.

70. The system of claim 68, wherein the access analyzer is connected to the Internet through a firewall.

ForcePoint LLC
EX-1009
Page232

71. The system of claim 68, further comprising at least one control unit operably connected to the Internet.

72. The system of claim 71, wherein the control unit is connected to the Internet through a firewall.

73. The system of claim 71, wherein said control unit and said access analyzer are implemented on the same computer.

74. The system of claim 71, wherein said control unit and said access analyzer are implemented on the same intranet.

75. The system of claim 71, wherein said control unit and said agent are implemented on the same computer.

76. The system of claim 71, wherein said control unit and said agent are implemented on the same intranet.

ForcePoint LLC
EX-1009
Page233

77. The system of claim 71, wherein said guardian agent implements a unified security policy which is a unification of the said systems original security policy and an security policy operable with at least said permission levels.

78. The system of claim 77, wherein said unified security policy implements the most restrictive rules.

79. The system of claim 77, wherein said unified security policy implements the most permissive rules.

80. The system of claim 77, wherein said security policy comprises at least one security rule.

81. The system of claim 80, wherein said security rule defines at least one response to said access attempt.

82. The system of claim 81, wherein said response comprises at least allowing an access permission.

83. The system of claim 51, wherein said response comprises at least denying an access.

ForcePoint LLC
EX-1009
Page234

84. The system of claim 81, wherein said response comprises at least notification of an access.

85. The system of claim 81, wherein said response is adaptive in accordance with at least said permission levels.

86. The system of claim 85, wherein said adaptive response is further determined in accordance with at least one activation threshold.

87. The system of claim 86, wherein said response comprises at least allowing an access, determined at least in accordance with a first activation threshold.

88. The system of claim 86, wherein said response comprises at least denying an access, determined at least in accordance with a first activation threshold.

89. The system of claim 86, wherein said response comprises at least notification of an access, determined at least in accordance with a second activation threshold.

90. The system of claim 69, further comprising a discovery unit operable to provide information to said control unit.

ForcePoint LLC
EX-1009
Page235

91. The computer network of claim 90, wherein said information includes data relative to at least a user of said network.

92. The computer network of claim 90, wherein said information includes data relative to at least a resource accessible through said network.

93. The computer network of claim 90, wherein gathering of said information is automatic.

94. The system of claim 68, wherein said agent responds to said access attempt in accordance with a result of comparing at least one permission level to at least one constant threshold.

95. The system of claim 94, wherein said agent permits access to said resource if the permission level exceeds a first threshold.

96. The system of claim 94, wherein said agent denies access to said resource if the permission level does not exceed a first threshold.

ForcePoint LLC
EX-1009
Page236

97. The system of claim 94, wherein said agent sends a notification of said access attempt to said control unit if the permission level is below a second threshold.

98. The system of claim 68, wherein said access attempt comprises at least one of: a user, a resource, a location, an access type, a time.

99. The system of claim 68, wherein said user may be one or more users.

100. The system of claim 68, wherein said resource may be one or more resources.

101. The system of claim 68, wherein said location may be one or more locations.

102. The system of claim 68, wherein said access type comprises at least one of the following; read, write, modify, execute, delete, rename, take ownership of, change permissions of, or create, said resource.

103. The system of claim 68, wherein said access attempt comprises at least a time scope.

104. The system of claim 68, wherein said event audit trail is derived from at least one access attempt.

ForcePoint LLC
EX-1009
Page237

105. The system of claim 104, wherein said event audit trail is provided to said access analyzer on a periodic basis.

106. The system of claim 104, wherein said event audit trail is periodically solicited by said access analyzer.

107. The system of claim 104, wherein said access attempt is ignored by said agent with respect to at least one access attempt.

108. The system of claim 68, wherein said permission levels are derived from at least likelihood estimates that a future access attempt should be permitted.

109. The system of claim 68, wherein said agent comprises at least one enforcement means for controlling access to said resources based on at least said permission levels.

110. The system of claim 68, wherein said permission levels are generated periodically based on a currently available event audit trail.

ForcePoint LLC
EX-1009
Page238

111. The system of claim 68, wherein said agent and said access analyzer reside on a single computer.

112. The system of claim 71, wherein said agent and said control unit reside on a single computer.

113. The system of claim 71, wherein said access analyzer and said control unit reside on a single computer.

114. The system of claim 71, wherein said agent, said access analyzer and said control unit reside on a single computer.

115. The system of claim 68, wherein said permission levels generated by said access analyzer are used for the purpose of updating at least an access control list of at least a system resource.

116. The system of claim 68, wherein said access analyzer provides permission level information to at least a system consulting unit.

117. The system of claim 116, wherein in response to said access attempt a query is sent to said system consulting unit.

ForcePoint LLC
EX-1009
Page239

118. The system of claim 117, wherein in response to said query asid system consulting unit provides a response corresponding at least to said permission levels.

119. The system of claim 68, wherein said agent is implemented as a proxy server.

120. A method for controlling access to resources available through the system, said method comprising:

establishing a first threshold value;

gathering an event audit trail comprising data corresponding to at least one access attempt;

periodically analyzing said event audit trail;

generating a permission level for accessing system resources from at least said event audit trail; and

controlling access to said resources based on said generated permission level.

121. A method as claimed in claim 120, wherein said generated permission level comprises adaptive permission levels generated by periodically automatically modifying said event audit trail to account for the most recent access attempt or access attempts.

ForcePoint LLC
EX-1009
Page240

122. The method of claim 120, wherein if said permission level exceeds said first threshold value, access to said resource is permitted.

123. The method of claim 120, wherein if said permission level does not exceed said first threshold value access to said resource is denied.

124. The method of claim 120, further comprising:

establishing a second threshold value; and

generating a notification if said permission level is below said second threshold value.

125. The method of claim 120, wherein said permission levels are derived from at least at least likelihood estimates that a future access attempt should be permitted.

126. The method of claim 125, wherein said likelihood calculations comprise a probability that a future access attempt should be permitted.

127. The method of claim 120, further comprising:

discovering information about users and said resources.

128. The method of claim 127, wherein said discovering occurs automatically.

ForcePoint LLC
EX-1009
Page241

129. A method for employing adaptive permissions for accessing system resources comprising of the steps of:

establishing a system security policy said security policy containing at least one adaptive security rule;

gathering an event audit trail of at least one access attempt;

periodically analyzing said event audit trail;

generating permission levels for accessing system resources from at least said event audit trail and said security policy; and

controlling access to said resources based on said generated permission levels.

130. The method of claim 129, wherein the security policy is formatted as a security policy table.

131. The method of claim 129, wherein said adaptive security rule comprises at least one threshold.

132. The method of claim 131, wherein if said permission level exceeds said first threshold value access to said resource is permitted.

133. The method of claim 131, wherein if said permission level does not exceed said first threshold value access to said resource is denied.

ForcePoint LLC
EX-1009
Page242

134. The method of claim 131, wherein if said permission level is below said second threshold value a notification is generated.

135. The method of claim 129, wherein said permission levels are derived from at least the likelihood estimates that a future access attempt should be permitted.

136. The method of claim 129, further comprising the step of discovering information about users and said resources.

137. The method of claim 136, wherein said step of discovering is automatic.

ForcePoint LLC
EX-1009
Page243

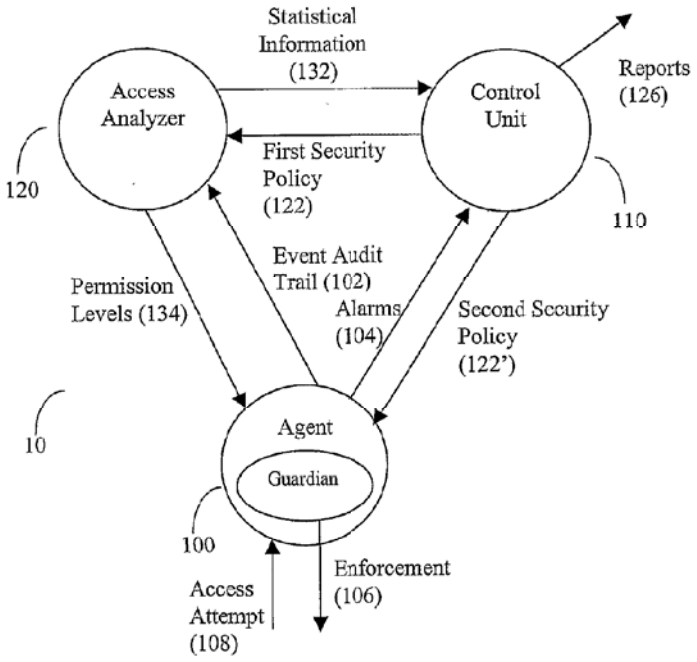

Statistical Information (132)
Access Analyzer
Control Unit
Reports (126)
First Security Policy (122)
120
110
Event Audit Trail (102)
Permission Levels (134)
Alarms (104)
Second Security Policy (122')
10
Agent
Guardian
100
Enforcement (106)
Access Attempt (108)

FIG.1

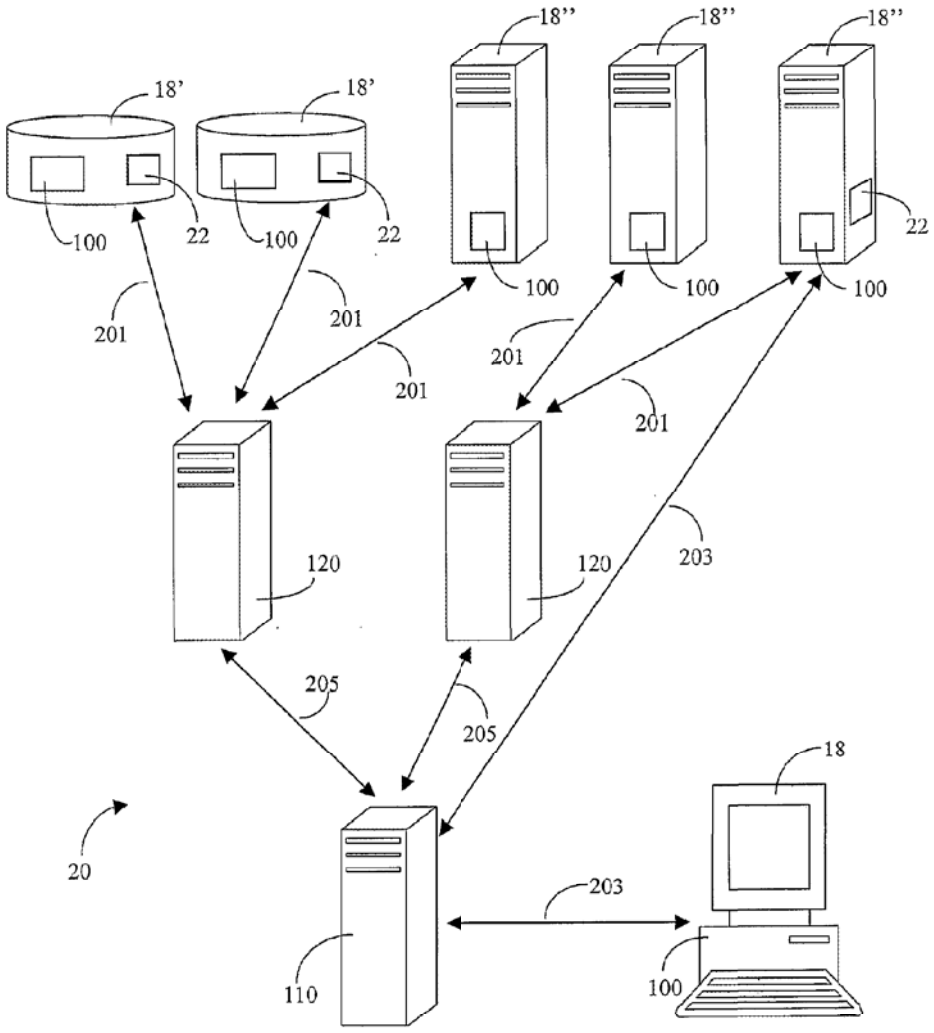

18''
18''
18''
18'
18'
22
22
100
100
100
100
100
100
201
201
201
201
201
201
120
120
203
205
205
18
20
203
110
100

FIG. 2

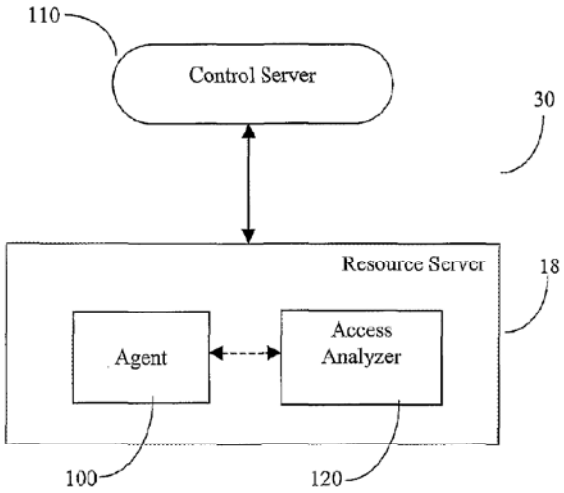


FIG. 3A

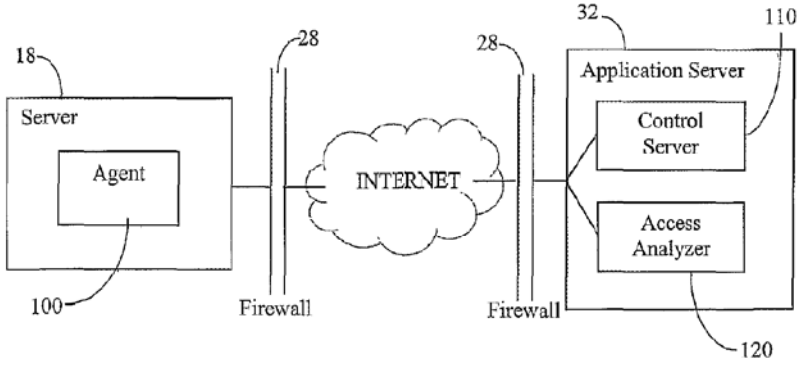


FIG. 3B

ForcePoint LLC
EX-1009
Page246

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
23 January 2003 (23.01.2003)

PCT

(10) International Publication Number
WO 03/007192 A1

(51) International Patent Classification[7]: G06F 17/30

(21) International Application Number: PCT/US02/22457

(22) International Filing Date: 10 July 2002 (10.07.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/304,270	10 July 2001 (10.07.2001)	US
60/313,793	20 August 2001 (20.08.2001)	US
10/054,307	22 January 2002 (22.01.2002)	US

(71) Applicant: **CORE SDI, INCORPORATED** [US/US]; 12th Floor, 44 Wall Street, New York, NY 10005 (US).

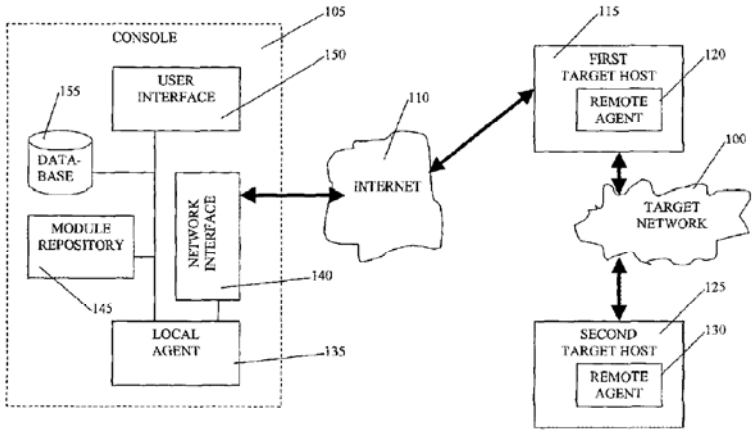
(72) Inventors: **CACERES, Maximiliano, Gerardo**; Freire 2140 10c Capital Federal, Ciudad Autonoma de Buenos Aires, CP 1428 (AR). **RICHARTE, Gerardo, Gabriel**; Pasaje Emilio Zola 5194, Ciudad Autonoma de Buenos Aires, CP 1425 (AR). **FRIEDMAN, Augustin, Azubel**; Juncal 2307 2do A, Ciudad Autonoma de Buenos Aires, CP 1425 (AR). **QUESADA, Ricardo**; Billinghurst 2444 6b, Ciudad Autonoma de Buenos Aires, CP 1425 (AR). **NOTARFRANCESCO, Luciano**; Av. Independencia 532, 6to C, Ciudad Autonoma de Buenos Aires (AR). **FRIEDRICHS, Oliver**; 442 Ninth Avenue, Menlo Park, CA 94025 (US). **BURRONI, Javier**; Salta 1363 13D, Ciudad Autonoma de Buenos Aires, CP 1137 (AR). **AJZENMAN, Gustavo**; Estado de Israel 4755 7 "A", Ciudad Autonoma de Buenos Aires (AR). **BECEDILLAS, Gabriel**; Jose Ingenieros 1468, Olivos, Provincia de Buenos Aires, CP 1636 (AR). **LEIDL, Bruce**; Lima 571, piso: 8 dept.: B, Ciudad Autonoma de Buenos Aires, C1073AAK (AR).

(74) Agents: **WISCHHUSEN, Carl, B.** et al.; Fitzpatrick, Cella, Harper & Scinto, 30 Rockefeller Plaza, New York, NY 10112-3801 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC,

[Continued on next page]

(54) Title: AUTOMATED COMPUTER SYSTEM SECURITY COMPROMISE



(57) Abstract: A system is provided for performing penetration testing of a target computer network by installing a remote agent (120) in the target computer network (100). The system includes a local agent (135) provided in a computer console (105) and configured to receive and execute commands. A user interface (150) is provided in the console and configured to send commands to and receive information from the local agent (135), process the information, and present the processed information. A database (155) is configured to store the information received from the local agent. A network interface (140) is connected to the local agent and configured to communicate with the remote agent installed in the target computer network (100) via a network (110). Security vulnerability exploitation modules are provided for execution by the local agent (135) and/or the remote agent (130).

WO 03/007192 A1

ForcePoint LLC
EX-1009
Page247

LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW.

(84) **Designated States** *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

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

ForcePoint LLC
EX-1009
Page248

TITLE

AUTOMATED COMPUTER SYSTEM SECURITY COMPROMISE

This application claims the benefit of U.S. Provisional Application No. 60/304,270, filed July 10, 2001, and U.S. Provisional Application No. 60/313,793, filed August 20, 2001.

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates generally to analyzing computer system security by compromising the security in a systematic manner.

Related Art

Computer systems that are connected to a computer network, such as the Internet, must employ security measures to prevent unauthorized users from accessing these systems. The security measures must be properly designed and implemented in order to prevent unauthorized access. However, it is difficult to evaluate the effectiveness of such security measures, particularly in view of the increasing sophistication of techniques used to gain unauthorized access to computer systems.

The effectiveness of the security measures of a computer system may be evaluated by performing a computer security audit in which various aspects of computer security are analyzed and evaluated.

ForcePoint LLC
EX-1009
Page249

The security audit may include a penetration test, which is a process by which a security auditor attempts to gain unauthorized access to the computer system.

Conventionally, a penetration test is performed using a multitude of *ad hoc* methods and tools, rather than according to a formalized standard or procedure. A typical penetration test includes the following stages:

1. Information gathering: The security auditor gathers technical details about the target system and information regarding the owner of the target system.

2. Information analysis and planning: The auditor analyzes the information to plan an overall approach by which to perform the penetration testing. This tends to be a difficult and time consuming task and requires experienced and knowledgeable personnel having a highly specialized skill base.

3. Vulnerability detection: The auditor searches the target system for security vulnerabilities based on the top-level plan developed in the information analysis and planning stage. Security vulnerabilities include, for example, system misconfigurations that enable an unauthorized user to gain access using a known series of steps.

The vulnerability search may be performed using an automated vulnerability scanner, which is a software package that determines whether certain known flaws may be used to gain unauthorized access to the target. Manual vulnerability scanning also may be performed to probe for common vulnerabilities that, for various reasons, may have been missed by an automated scanner. However, such vulnerability scanning techniques merely list the vulnerabilities, rather than actually attempt to exploit them.

ForcePoint LLC
EX-1009
Page250

The automated and manual vulnerability searches may be supplemented by research performed by the security auditor to determine previously unknown vulnerabilities. Such research typically is performed using a copy (also called a mirror) of the software application being probed and/or the associated hardware.

4. Compromising and accessing the target system: The auditor attempts to compromise the target system based on the results of the vulnerability detection stage using publicly available or custom-developed programs.

Publicly available programs designed to exploit system vulnerabilities tend to be unreliable and require testing and customization before use. In general, exploiting detected vulnerabilities, regardless of the tools being used, requires experienced and knowledgeable personnel having a highly specialized skill base. In addition, a considerable laboratory infrastructure may be required to develop and test vulnerability exploitation tools, particularly when the target system employs a number of different operating system platforms.

5. Analysis and reporting: This stage includes consolidating and presenting the information obtained during the previous stages and developing recommendations for remedying the security vulnerabilities identified during the penetration test. Manually maintaining a record of all of the actions taken and information gathered during testing is extremely time consuming and prone to error. Moreover, the preparation of complete and accurate records is subject to the discipline of the personnel conducting the test.

6. Clean up: The compromising and accessing stage typically results in significant changes being made to the target system. In the clean up stage, the auditor returns the system to its original

ForcePoint LLC
EX-1009
Page251

configuration. To perform a successful clean up, a detailed and exact list of all actions performed during testing must be maintained, yet there are only rudimentary tools available for maintaining such information.

In view of the shortcomings discussed above, there is a need for a system and method for performing an automated penetration test employing computer system compromise that takes an entirely fresh approach and overcomes the drawbacks of the conventional techniques.

SUMMARY OF THE INVENTION

The present invention generally provides a novel system, method, and computer code for performing automated penetration testing for analyzing computer system security by compromising the security in a systematic manner.

One aspect of the present invention provides a system, method, and computer code for performing penetration testing of a target computer network by installing a remote agent in the target computer network. A local agent is provided in a console and configured to receive and execute commands. A user interface provided in the console and configured to send commands to and receive information from the local agent, process the information, and present the processed information. A database is configured to store the information received from the local agent. A network interface is connected to the local agent and configured to communicate via a network with the remote agent installed in the target computer network. Security vulnerability exploitation modules are provided for execution by the local agent and/or the remote agent.

Embodiments of this aspect may include one or more of the following features. The user interface may enable a user to select one of the modules and initiate execution of the selected module on

either the local agent or the remote agent. The user interface may provide a graphical representation of the target computer network.

Another aspect of the present invention provides an agent for use in a system for performing penetration testing of a target computer network. The agent includes a proxy server configured to receive and execute system calls received via a network and a virtual machine configured to execute scripting language instructions received via the network.

Embodiments of this aspect may include one or more of the following features. The agent may include an execution engine configured to control the proxy server and the virtual machine. The system calls and the scripting language instructions may be routed to the proxy server and the virtual machine, respectively, by the execution engine. The agent may include a remote procedure call module configured to receive commands from the network formatted in a remote procedure call protocol and pass the commands to the execution engine.

Another aspect of the present invention provides an agent including a proxy server configured to receive and execute system calls received via a network and a virtual machine configured to execute scripting language instructions received via the network. The agent further includes a secure communication module configured to provide secure communication between the virtual machine and the network. An execution engine is configured to control the proxy server and the virtual machine. The system calls and the scripting language instructions are routed to the proxy server and the virtual machine, respectively, by the execution engine. A remote procedure call module is configured to receive commands via the network formatted in a remote procedure call protocol and pass the commands to the execution engine. A second secure communication module

ForcePoint LLC
EX-1009
Page253

is configured to provide secure communication between the remote procedure call module and the network.

Another aspect of the present invention provides a system, method and computer code for performing penetration testing of a target network in which a first module is executed in a console having a user interface. The first module is configured to exploit a security vulnerability in a first target host of the target network. A first remote agent is installed in the first target host. The first remote agent is configured to communicate with the console and a second remote agent. A second module is executed in the first remote agent. The second module is configured to exploit a security vulnerability in a second target host of the target network.

Embodiments of this aspect may include the feature of installing a second remote agent in the second target host of the target network, the second remote agent being configured to communicate with the first remote agent.

Another aspect of the present invention provides a system, method, and computer code for performing penetration testing of a target network. A first module is executed to exploit a security vulnerability of a first target host of the target network. A first remote agent is installed in the first target host as a result of exploiting the security vulnerability of the first target host. A system call is sent to the first remote agent via a network. The system call is executed in the first target host using a proxycall server of the first remote agent to exploit a security vulnerability of a second target host.

In another aspect, a first module is executed to exploit a security vulnerability of a first target host of the target network. A first remote agent is installed in the first target host as a result of

ForcePoint LLC
EX-1009
Page254

exploiting the security vulnerability of the first target host. A second module that generates a system call is executed in the first remote agent. The system call is executed in the first target host to exploit a security vulnerability of a second target host.

In another aspect, a first module is executed to exploit a security vulnerability of a first target host of the target network. A first remote agent is installed in the first target host as a result of exploiting the security vulnerability of the first target host. A second module is executed in the first remote agent that generates a system call. A second remote agent is installed in the second target host as a result of exploiting a security vulnerability of the second target host. The system call generated by the second module is sent to the second remote agent via a network. The system call is executed in the second target host using a proxycall server of the second remote agent.

In another aspect, a first module is executed to exploit a security vulnerability of a first target host of the target network. A first remote agent is installed in the first target host as a result of exploiting the security vulnerability of the first target host. A second remote agent is installed in the second target host as a result of exploiting a security vulnerability of the second target host. A system call is sent to the first remote agent and is sent from the first remote agent to the second remote agent. The system call is executed in the second target host using a proxycall server of the second remote agent.

In another aspect, a first remote agent is installed in the first target host. The first remote agent has a proxy server configured to receive and execute system calls. A system call received via a network is executed in the first remote agent. A second remote agent is installed in the first target host. The second remote agent has a proxy server configured to receive and execute system calls

ForcePoint LLC
EX-1009
Page255

and a virtual machine configured to execute scripting language instructions. A scripting language instruction or a system call received via the network is executed in the second remote agent.

These and other objects, features and advantages will be apparent from the following description of the preferred embodiments of the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be more readily understood from a detailed description of the preferred embodiments taken in conjunction with the following figures.

Fig. 1 is a block diagram of a system for performing automated penetration testing of a target network in accordance with the present invention.

Fig. 2 is a block diagram of a console for automated penetration testing connected through the Internet to a first target host.

Fig. 3 is a block diagram of the console connected through the Internet to a first target host, which is connected through a target network to a second target host.

Fig. 4 is a block diagram showing components of the console.

Fig. 5 is a block diagram of a level 0 agent.

Fig. 6 is a block diagram of a level 1 agent.

Fig. 7 is a block diagram of a level 2 agent connected to upstream and downstream agents by two different networks.

ForcePoint LLC
EX-1009
Page256

Fig. 8 is a block diagram of a level 2 agent connected to upstream and downstream agents by a single network.

Fig. 9 is a block diagram of a level 3 agent.

Fig. 10 is a block diagram of a configuration in which a module is executed by the local agent in the console and system calls are executed in the operating system of the console.

Fig. 11 is a block diagram of a configuration in which a module is executed by the local agent in the console and system calls are executed in the operating system of the first target host.

Fig. 12 is a block diagram of a configuration in which a module is executed by the remote agent in the first target host and system calls are executed in the operating system of the first target host.

Fig. 13 is a block diagram of a configuration in which a module is executed by the remote agent in the first target host and system calls are executed in the operating system of the second target host.

Fig. 14 is a block diagram of a configuration in which a module is executed by the local agent in the console and system calls are executed in the operating system of the second target host.

Fig. 15 is a basic graphical user interface display screen presented by the user interface of the console.

Fig. 16 is a display screen following the running a Network Discovery Module.

Fig. 17 is a display screen following initiation of the Network Discovery module for a second time.

Fig. 18 is a display screen showing the results of the second running of the Network Discovery module and the running of the Remote Procedure Call (RPC) Mapper module.

ForcePoint LLC
EX-1009
Page257

Fig. 19 is a display screen in which the entities in the model of the target network can be examined and modified using an entity editor.

Fig. 20 is a display screen following two executions of the General Exploit module and the installation of a level 0 agent in a target host.

Fig. 21 is a display screen showing a window in which a Python console is running using the level 0 agent as a source.

Fig. 22 is a display screen following the execution of a port scanning module using the level 0 agent as a source.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

According to the present invention, as shown in Fig. 1, an automated penetration test is performed to identify, analyze, exploit, and document security vulnerabilities in a target network 100. The penetration test is executed by a console 105 that may be, for example, a personal computer running Microsoft Windows 2000 Professional, Server, or Advanced Server operating systems. The target network 100 may be connected to a network, such as for example the Internet 110. In the case of such example, the console also would be connected to the Internet 110 and would gain access to the target network 100 through the Internet 110.

The target network 100 has a first target host 115, e.g., a firewall. The firewall is a security device that typically is the only host in the target network that is connected directly to the Internet. It handles Internet traffic to and from the target network and serves to protect the target network from unauthorized access. The target network 100 has a number of other hosts connected to it – all of which could be the eventual targets of the penetration test.

ForcePoint LLC
EX-1009
Page258

The console 105, shown in Fig. 2, compromises the security measures protecting the first target host 115 by executing a series of modules. The modules may be selected and initiated by the user. Alternatively, the console may execute a predetermined sequence of modules or may determine a sequence of modules to be executed based on the information gathered during the penetration testing.

In the initial stage, typically, modules are executed to gather information about the first target host 115. For example, the console 105 may execute a port scanner that analyzes the ports of the first target host 115 and determines all of the services that are being run, such as an Internet web server, an email server, a finger service, etc. Further information might be acquired by running modules designed to exploit the services identified by the port scanner. For example, if the first target host 115 is running a finger service, then that service will be targeted by a module to determine software version, user names, etc. As a further example of an information gathering module, a network discovery module may be used to determine the number of hosts in the target network 100 and the Internet Protocol (IP) address of each host.

Following execution of the information gathering modules, the console executes modules, called exploits, to exploit security vulnerabilities in the first target host 115 based on the information that has been retrieved. For example, information may be obtained regarding a firewall operating system being run on the first target host 115, such as the software brand and revision number. Based on this information, the console 105 executes an exploit that has been written to take advantage of security vulnerabilities for that particular firewall.

Once a service running on the first target host 115 has been compromised, the console 105 installs a remote agent 120 on the first target host. The remote agent 120 is a program that operates on the

first target host 115 to perform a number of functions, such as receiving and executing control commands and modules from the console 105 and sending back information to the console 105.

As shown in Fig. 3, once the remote agent 120 has been installed on the first target host 115, it is used by the console 105 to gain access to the target network 100 and compromise the security of the other hosts that make up the target network 100, such as the second target host 125. For example, the remote agent 120 on the first target host 115 executes exploits, such as those discussed above, or system calls received from the console 105 to gather information and exploit other security vulnerabilities in the target network 100. To hosts connected to the target network 100, such commands or queries appear to originate from the first target host 115 and therefore may be more readily accepted. This is particularly true for networks that employ hierarchies of trust, which means that commands from known, i.e., trusted, sources are subject to less stringent security measures than commands from unknown sources.

Once the security of the second target host 125 has been compromised, the remote agent 120 in the first target host 115 installs a remote agent 130 in the second target host 125. Each of the installed agents 120 and 130 sends and receives modules, commands, and data from other installed agents, which is referred to as chaining. For example, the agent 130 in the second target host 125 receives modules and commands from the agent 120 in the first target host 115, which, in turn, receives the modules and commands from the console. The agent 130 in the second target host 125 also sends data back to the agent 120 in the first target host 115, which, in turn, sends the data back to the console 105.

The agent 130 in the second target host 125 executes the modules received from the upstream agents to gather information from and exploit security vulnerabilities in a third target host, in a

ForcePoint LLC
EX-1009
Page260

manner similar to that discussed above. Once the security measures have been compromised, the agent in the second target host installs an agent in the third target host. The penetration of the target network may continue in this manner until all of the target hosts have been compromised or until the final target of the penetration testing has been compromised.

The term "exploiting security vulnerabilities", as used herein, is a broad concept that includes any means for gaining access to and/or obtaining information from a target host. The concept includes, without limitation, the execution of modules that are designed to take advantage of specific security vulnerabilities that have been identified in a target host. For example, the first target host 115 may have been misconfigured by the owner in a manner that is detectable and allows installation of a remote agent 120. This concept also includes, without limitation, the execution of information gathering modules, such as port scanners and network discovery modules. This concept further includes, without limitation, the exploitation of security vulnerabilities of a target host that result from the compromise of other target hosts. For example, once a remote agent 120 has been installed in the first target host 115, it may be possible to gather information and install remote agents on other target hosts due to hierarchies of trust within the target network. This concept further includes, without limitation, obtaining access to a target host by virtue of a lack of security features or measures.

As shown in Fig. 4, a local agent 135 installed in the console 105 communicates with and controls the remote agents 120 and 130 through a network interface 140, such as a network interface card. The console 105 provides a module repository 145 to store modules that perform various functions related to system penetration and compromise. During the penetration test, these modules are sent to the local agent 135 in the console 105 or a remote agent 120 or 130 in a target host 115 or 125 to

ForcePoint LLC
EX-1009
Page261

be executed. For example, a module that performs packet sniffing may be sent to and executed on the remote agent 120 in order to monitor data packets passing through the first target host 115.

The console 105 also provides a user interface 150 that enables the user to control and monitor the performance of the penetration test. The user interface 150 may be a graphical user interface that displays a representation of the target network 100. The user interface 150 also reports the results of the activities performed by the agents installed in the target network.

A database 155 in the console 105 stores information sent back by remote agents 120 and 130. The database 155 could be a standard flat file, for example a text file, although a more complex structure, such as a relational database, also may be used. In addition, data may be exported to an external relational database. Each structured element of information is represented as an object in the database 155. The database 155 can accommodate multiple distributed instances of the same object that can be independently updated. The database 155 synchronizes the multiple instances of the objects to allow revision of the instances with the most recent data.

The information that is stored as objects in the console database 155 includes an activity log of all actions performed by the remote agents 120 and 130, such as executing modules, port scanning the host, passing modules to downstream remote agents, and installing remote agents in other hosts. The information also includes configuration data relating to the deployment of the remote agents, such as identification of the hosts in which the remote agents are installed and the possible communication channels that might be used to connect to them. The information also includes all of the data generated by execution of commands and modules by the remote agents, such as known hosts in the target network, operating systems, open ports, user accounts, cracked passwords, etc. The remote agents may store a subset of the database to provide caching of module data and to

ForcePoint LLC
EX-1009
Page262

make such data available to downstream modules. The cached data is eventually sent back to the console database.

Another type of object that can be managed by the database 155 is referred to as an entity, which is an object that represents a physical component in the target network 100, such as a host. Entities contain information similar to that discussed above, but also may have the capability to perform certain functions. For example, an entity may be capable of serializing and deserializing itself to and from XML markup language for transfer in and out of the database. This allows information produced by modules executed by remote agents to be shared between agents and with the console.

The data stored in the console database 155 is used for a number of purposes, such as to analyze the security vulnerabilities of the target network 100 and plan the exploitation of these vulnerabilities to penetrate further into the target network 100. The data also allows the console to reverse any changes made to the target network 100 during the penetration testing. In addition, the data allows the client for whom the penetration testing is performed to have a complete picture of actions performed during the testing. For example, the data may be exported into a relational database, as mentioned above, or may be used to generate reports using XML.

In general terms, an agent, such as the local and remote agents discussed above, is a program that is controlled by or used as a proxy by another program. The controlling program may be resident on the same host as the agent, as in the case of the user interface that controls the local agent on the console. Alternatively, the controlling program may be on a separate host, as in the case of the local agent on the console that controls the remote agent on the first target host. Hence, in this example, the local agent in the console serves both as an agent and as a controlling program.

Fig. 5 shows an example of a basic agent, which is referred to as a level 0 agent 205, running on a host 210 that is connected to a network 215. The level 0 agent 205 communicates through the network 215 to a controlling program running on an upstream agent, such as the local agent 135 in the console 105 or an intermediate agent positioned between the console and the level 0 agent. The level 0 agent 205 provides a syscall proxy server 220 that enables it to act as a proxy for the controlling program. This configuration is referred to as a system call proxy or proxycall configuration.

For example, referring again to Fig. 4, a level 0 agent 205 may be installed as the remote agent 120 on the first target host 115 and may be controlled by a module executed by the local agent 135 in the console 105 such as, for example, a remote exploit. The module includes commands that result in system calls ("syscalls"), which are instructions that access the operating system of the host. Rather than executing the syscall on the console 105, the syscalls are sent by a proxy client in the local agent 135 to the syscall proxy server 220 in the remote agent 120. The syscall proxy server 220 executes the syscalls on the first target host 115. Thus, syscalls are executed on the host of the remote agent 120, e.g., the first target host 115, rather than the host of the local agent 135, e.g., the console 105.

By acting as a syscall proxy, the remote agent 120 allows the local agent 135 to execute commands as if the local agent 135 were resident on the first target host 115. The syscall proxy configuration allows large, complex programs, such as the automated penetration testing program resident on the console 105, to execute system calls on a target host without actually being resident on the target host. Only the relatively small agent needs to be installed on the target host.

ForcePoint LLC
EX-1009
Page264

The level 0 agent 205 is a relatively simple program and may be about 100 bytes in size or smaller. Due to its small size, the level 0 agent 205 is typically the first type of agent to be installed in the target host during penetration testing. It may be used to gather information and/or exploit vulnerabilities in the target host to enable further penetration of the target host. Thus, the level 0 agent helps open the way for the installation of more complex agents, as discussed below.

The level 0 agent is often installed directly as the result of exploiting a security vulnerability, e.g., during exploitation of a buffer overflow or user-supplied format string vulnerability in an application running on the target host. The installed agent runs in the process space of the compromised application, e.g., an Internet web server or email server, and opens a socket to provide an unencrypted communication channel to the console or upstream agent. The syscall proxy server 220 in the level 0 agent 205 can execute one syscall at a time.

Fig. 6 shows an example of a more complex agent, which is referred to as a level 1 agent 225. The level 1 agent 225, like the level 0 agent 205, provides a syscall proxy server 220 to execute commands received from the console 105 or an upstream agent. The level 1 agent 225 can spawn separate syscall proxy servers 220 for corresponding syscalls received from the upstream agent, so several syscalls can be executed simultaneously. The level 1 agent 225 also can allocate memory and other resources within the host. Hence, the level 1 agent 225 can run in its own process space rather than relying solely on the process space of a compromised application. The ability to run in its own process space gives the level 1 agent 225 more autonomy and control over its own lifetime and more control over the resources of the host 210.

The level 1 agent 225 also provides a secure communication channel 230 with authentication and encryption of data. Data transmitted by the level 1 agent 225 back to the console 105 or upstream

ForcePoint LLC
EX-1009
Page265

agent is encrypted to ensure that information about the compromised system cannot be detected by third parties that may be monitoring network communications. The data is authenticated, e.g., using a digital signature, so that only the console that created the agent can communicate with it. This prevents different users who may be running the security compromise system from communicating with each other's agents.

Fig. 7 shows an example of a more complex agent, which is referred to as a level 2 agent 305. The level 2 agent 305, like the level 1 agent 225, provides a syscall proxy server 220 and secure communication 230 capability. The syscall proxy server 220 allows the level 2 agent to execute syscalls received from a module running on an upstream agent in a manner similar to the level 0 and level 1 agents. For example, a module running on the local agent 135 in the console 105 such as, for example, a remote exploit, may generate system calls that are intended to be executed in the operating system of the target host, rather than the console. Such syscalls are sent to the syscall proxy server 220 of the remote agent in the target host to be executed.

The level 2 agent 305 also provides a virtual machine 310 that can execute modules written in high-level, platform-independent, scripting languages, such as Python, Perl, Java, and Tcl. A module, in general, is an element of program code containing an individual operation or group of operations that are to be executed. Most of the modules used in the automated penetration test are designed to exploit security vulnerabilities in or gather information from a target host. Such modules may be obtained in a variety of ways. For example, laboratory testing may be performed on widely used software products to determine possible security vulnerabilities and how to exploit them. A module is then written for each software product based on the results of this testing. As a further example, such modules are commonly written and made available on the Internet by

ForcePoint LLC
EX-1009
Page266

members of the on-line security research community or by hackers or crackers, i.e., persons who attempt to gain unauthorized access to computer systems.

One advantage of executing modules in a virtual machine 310 is that it allows the level 2 agent 305 to perform computationally-intensive operations on the target host 210 without receiving a continuous flow of instructions, as in the case of syscall proxying. Indeed, syscall proxying may not be feasible for computationally-intensive operations because of the delays in transmitting instructions that result from network latency. Thus, the virtual machine 310 of the level 2 agent 305 can take greater advantage of the processor and other resources of the target host 210.

Another advantage of executing modules in a virtual machine 310 is that the modules need not be included as an integral part of the agent, but instead may be transferred to the agent on demand. This obviates the need to port the modules to the target host 210 operating system. Thus, modules can be run in a variety of target hosts without being rewritten for each different target host operating system. Moreover, the size of the agent is substantially reduced. In addition, because there are many widely available programs and modules written in standard scripting languages, the amount of custom programming required to create modules to execute on the target host is reduced.

The virtual machine 310 can communicate with, control, and install modules in downstream agents by opening a network connection to the target host 210 to access a network 315. The communication with the downstream agents may be established by a secure communication module 230 that forms a secure communication channel with data encryption and authentication. The virtual machine 310 can also handle native calls, which are syscalls that are intended to be executed in the operating system of the target host 210 on which the virtual machine 310 is installed. For example, an information gathering module may execute a syscall in the operating

ForcePoint LLC
EX-1009
Page267

system of the target host 210 on which the virtual machine 310 is installed in order to access the Internet through a network interface.

In the example shown in Fig. 7, the network 315 accessed by the virtual machine 310 is separate from the network 215 used to communicate with the console 105 or upstream agent. The network 315 accessed by the virtual machine 310 may be a target network 100, such as shown in Figs. 1 and 3, that cannot be reached from an outside host. Alternatively, as shown in Fig. 8, the network 320 accessed by the virtual machine 310 may be the same network 320 used to communicate with the console 105 or upstream agent. For example, as shown in Fig. 1, both the second and third target hosts are connected to the target network, and a virtual machine in an agent on the second target host may communicate with an agent installed on the third target host.

The virtual machine 310 and the syscall proxy server 220 in the level 2 agent 305 are controlled by an execution engine 325 that determines which modules are executed and how they are executed. For example, the execution engine 325 provides task control capability to start, pause, resume, and cancel module execution. The execution engine 325 also controls the execution of native modules 330, which are modules written to be executed directly in the operating system of the target host 210.

The execution engine 325 provides transparent multitasking capabilities by taking advantage of either the multi-threading or multi-processing capabilities of the underlying operating system to enable concurrent execution of modules and syscalls by spawning new threads or processes as necessary. Task-specific storage is provided to enable an executing module to access context information. For example, upon initialization, the execution engine may set up synchronization

ForcePoint LLC
EX-1009
Page268

objects (known as mutexes) and task-specific storage for module context using, e.g., the ThreadLocalStorage routine in Win32 or the thread-specific routine in PTHREADS.

The execution engine 325 is isolated in certain respects from the rest of the system architecture, aside from the modules that are controlled by it. For example, the execution engine 325 operates independently of the proxycall architecture. In addition, necessary parameters for the modules to execute on the agent are provided by the module context, rather than by the execution engine 325.

The module context provides all of the information necessary for a specific execution instance (the execution instance being, e.g., a task or executing module). Such information includes the module parameters, designation of remote or local execution, description of the peer (for remote execution), etc. The module context is available from any subsystem of the agent that is involved in the module execution, e.g., a proxycall client. The module accesses the module context from the agent during execution of the module.

The level 2 agent receives commands and instructions through a remote procedure call (RPC) component 327, which then passes the received data to the execution engine 325. The RPC component 327 employs a standard protocol to control communication with the console or upstream agent using a predefined data packet format. To implement this protocol, the console uses an internal representation of all the deployed agents that includes all of the properties and capabilities for establishing a communication channels with the deployed agents.

Communications between the console and deployed level 1 and level 2 agents are established as follows. First, the console establishes an internal connection with the internal representation of the remote agent with which it is attempting to communicate. The internal representation of the

ForcePoint LLC
EX-1009
Page269

remote agent uses its communications component to establish a communication channel across the network with the remote level 1 or level 2 agent. The communication channel then is created using a component of the system that can multiplex communications from several modules into one or many communications channels with a remote agent.

The RPC component 327 implements a bi-directional communication mechanism using the communications channel described above as an underlying transport mechanism. The RPC component implements and executes commands that control the execution of modules on the remote agents as well as the transfer of information between agents. For example, the RPC component is used to start, pause and stop module execution. The RPC component also is used to transfer modules to remote agents, retrieve module execution results from remote agents, and control the configuration and status of remote agents.

Fig. 9 shows an example of a level 3 agent 335, which is similar to the level 2 agent 305 shown in Fig. 8, but includes a user interface 340 through which the level 3 agent 335 is controlled. The user interface 340 allows the level 3 agent 335 to function as the local agent 135 component of the console 105, as shown in Fig. 2. The level 3 agent 335, like the level 2 agent 305, includes secure communication capability 230, a virtual machine 310, a syscall proxy server 220, native module capability 330, an RPC send/dispatch component 327, and an execution engine 325.

The proxycall and virtual machine capabilities discussed above enable the agents to execute modules in a number of different configurations. Fig. 10 shows a configuration in which a module 400 is executed by the local agent 135 in the console 105. System calls generated by the module 400 are handled by the local agent 135 as native syscalls and are executed in the operating system 405 of the console 105. For example, when a module 400 executes a command for querying the

status of a port on a target host 115, a system call is made to the operating system 405 of the console 105 in order to access the network interface 140 and transmit the request to the target host 115 through the Internet 110.

Another configuration in which a module is executed by the local agent 135 in the console 105 is shown in Fig. 11. In this configuration, however, system calls generated by the module 400 are transmitted to a remote agent 120 installed in the first target host 115. The system calls are handled by the syscall proxy server of the remote agent 120 and are executed in the operating system 410 of the first target host 115. For example, when a module 400 executes a command for retrieving a file from the second target host 125 through the target network 100, the syscall generated by this command is executed in the operating system 410 of the first target host 115. From the perspective of the second target host 125, the command appears to have been executed on the first target host 115 and therefore is more likely to be trusted and implemented by the second target host 125.

Fig. 12 shows a configuration in which a module 400 is executed by a remote agent 120 in a target host, e.g., the first target host 115. System calls generated by the module 400 are handled by the remote agent 120 as native syscalls and are executed in the operating system of the first target host 410. This configuration allows the module 400 to take advantage of the resources of the first target host 115. In addition, it enables the module 400 to run without generating network traffic between the console 105 and the target network 100.

Another configuration in which a module is executed by a remote agent in a target host is shown in Fig. 13. In this configuration, however, the system calls generated by the module 400 are transmitted to a remote agent 130 installed in another target host, e.g., the second target host 125.

ForcePoint LLC
EX-1009
Page271

System calls generated by the module 400 are handled by the syscall proxy server of the remote agent 130 and are executed in the operating system 415 of the second target host 125.

As above, this configuration allows the module to take advantage of the resources of the first target host 115, and enables the module to run without generating network traffic between the console 105 and the target network 100. Another advantage of this configuration is that, from the perspective of the other hosts in the target network 100, the module 400 commands appear to originate from the second target host 125. Consequently, such commands are more likely to be trusted and implemented by the other hosts of the target network. In other words, using the remote agent 130, the system effectively assumes the identity of the compromised host and takes advantage of the privileges of the compromised host within the target network.

Fig. 14 shows another configuration in which a module is executed by the local agent in the console. In this configuration, however, system calls generated by the module 400 are transmitted to a remote agent 120 installed in the first target host 115, which in turn passes the system calls to a remote agent 130 installed in the second target host 125. The system calls are handled by the syscall proxy server of the remote agent 130 in the second target host 125 and are executed in the operating system 415 of the second target host 125. As in the case above, from the perspective of the other hosts in the target network, the module commands appear to originate from the second target host. Consequently, such commands are more likely to be trusted and implemented by the other hosts of the target network.

As discussed above, the local agent 135 in the console 105 provides a user interface 150. As shown in Fig. 15, the user interface presents to the user a Windows-based graphical user interface (GUI) screen. The screen has several component windows that present information in various

formats and allow control to be executed over various aspects of the penetration testing. Of course, the components of the screen may be moved and sized based on the user's preferences.

The visibility view, in the central portion of the screen, presents a graphical representation, i.e., a model, of the target network. In this example, the graphical representation is in the form of a hierarchical listing of entities, i.e., a tree listing. Each entity represents a physical component of the target network, e.g., a host. Initially, the visibility view shows only two entities, the local host (e.g., the console) and the local agent that is running on the local host. As further discussed below, as each stage of the penetration testing is performed, the model of the target network is updated to reflect newly gained information.

Tabs are arranged on the left side of the screen to organize the available functions, modules, and tools offered by the program. Clicking on a tab causes it to move to the top of the window and display the available modules under that category. For example, the Remote Exploits tab provides various modules that may be run on the local agent to exploit security vulnerabilities in a target network without being located in the target network. An executed modules window is provided on the top right side of the screen to display a history of the modules that have been executed during the penetration testing and their status. A module output window is provided on the bottom right side of the screen to display the output of the modules.

Fig. 16 shows an example of how the user interface screen appears after the Network Discovery module has been run. The Network Discovery module, which is grouped under the Information Gathering tab, is typically run early in the penetration testing to determine the topology of the target network. The executed modules window shows that the Network Discovery module has been run and provides the start and finish times and the current status, which in this case indicates that the

execution of the module is finished. The module output window provides a table of host names, Internet protocol (IP) addresses, operating system (OS), and other information output by the module.

Following the execution of the Network Discovery module, the visibility view displays a graphical representation of the target network in the form of a tree listing. The tree shows that the local host on the console (local host) has a local agent that is connected to a first target host having IP address 192.168.66.0/24. The first target host, in turn, is connected to other hosts within the target network. The entities in the tree listing may be collapsed or expanded according to hierarchical level by clicking on the arrow symbols on the left-hand side of the entities. As each module is run during the penetration testing, the visibility view is automatically updated to show a more complete model of the target network.

Fig. 17 shows an example of the screen display for initiating the execution of the Network Discovery module for a second time. The user double clicks on the Network Discovery module tab, and the screen presents a module parameter entry box. The user is prompted to enter the IP address of the target network, which is typically known by the user prior to performing penetration testing. For instance, the user may determine the IP address from a domain name server lookup of the domain name of the target host. In this example, the Network Discovery module is targeting the host with IP address 192.168.22.0/24. In the visibility view, the previously targeted host (192.168.66.0/24) has been collapsed to a single top-level entry in the tree listing. As an alternative to entering the target host IP address, the module tab may be dragged and dropped onto the target host in the tree listing for which information is to be gathered.

ForcePoint LLC
EX-1009
Page274

As discussed above, modules may be run on local or remote agents in a variety of configurations. The system calls generated by the modules are executed on the default source host, which in the example of Fig. 17 is the local agent. The default source is either the most recently used source or a source selected in the visibility view prior to execution of the module. For instance, if a module is executed using the local agent as a source host, the local agent remains the default source until a new source is selected. Thus, double clicking on the Network Discovery tab results in that module being run with on the local agent and gathering information from the target host entered in the module parameter entry box.

As shown in Fig. 18, the output of the Network Discovery module is presented in the visibility view as a list of connected hosts under the target host (192.168.22.0/24). The executed module window indicates that the Network Discovery module has been run a second time. In addition, the Remote Procedure Call (RPC) Mapper module has been run, which attempts to identify all of the ports in the target host that are available to receive RPC commands. The RPC Mapper was initiated by dragging and dropping it onto the target host, which in this case is 192.168.22.2. The output of the RPC Mapper is shown in the module output window in the log/debug format, which includes information to enable the user to track and rectify errors in the module. The output identifies the RPC ports and their associated protocols.

As shown in Fig. 19, the entities in the model of the target network can be examined and modified using an entity editor. Each entity has properties that reflect aspects of the information gathered on the corresponding physical component in the target network. For example, a host entity has properties such as system architecture (arch), operating system, interfaces, ports, etc. The host entity also has an agent property that lists the agents running on that host.

ForcePoint LLC
EX-1009
Page275

The properties of each entity are updated automatically as modules are run during the penetration testing. For example, the entity corresponding to host 192.168.22.2 has a listing of ports available for RPC service. This listing is the result of running the RPC Mapper module directed toward this host, as described above. In certain instances, different modules may result in different property states, for example, two different results regarding the operating system being run on a particular host. In such a case, the program will prompt the user to resolve the apparent conflict based on experience or outside knowledge.

Even in the absence of a conflict, the state of these properties may be changed by the user based on experience and knowledge gained from other sources. For example, the user may know, based on experience, that a host having a particular architecture, e.g., Sparc Version 8, may run a particular operating system, e.g., Solaris. In such a case, the user edits the host entity and changes the operating system property from unknown to Solaris. This allows the user to build a more complete and accurate model of the target network.

Once sufficient information has been gathered regarding the target network, modules may be executed to exploit security vulnerabilities in the target network. As shown in Fig. 20, such modules are grouped under the Local Exploits and Remote Exploits tabs on the user interface screen depending upon whether they run within the target host (i.e., locally) or remotely. In this example, the executed module window shows that a module called "General Exploit" has been run twice. The first execution of the module was unsuccessful, as indicated by the icon to the left of the module name and by the term "aborted" in the status column. The second execution was successfully completed. In addition to the modules grouped under the exploit tabs, there are a series of modules under the Agents tab that relate to the installation and execution of remote agents

ForcePoint LLC
EX-1009
Page276

in the target network. In this example, a level 0 agent has been installed in the host with address 192.168.22.6.

The installation of the level 0 agent may be accomplished by clicking on two entries under the Agents tab. First, a level 0 agent service is initiated, which is a process that runs on the target host. As discussed above, the level 0 agent service may share the process space of an existing service on the target host, such as an email server. Then, a level 0 agent is installed and executed in the level 0 agent service. Alternatively, the program may present a single entry under the Agents tab that performs all of the steps required to install and execute the level 0 agent.

Once the level 0 agent has been installed, the user can begin to take advantage of the agent by executing modules on it. The user sets the level 0 agent to be the source for the execution of modules by right clicking on the agent and selecting "Set as source". As shown in Fig. 21, the level 0 agent may be displayed as highlighted or bold in the visibility view to indicate to the user that it is the current default source. All subsequent modules are run on the level 0 agent source until the default is changed.

In this example, the Python Console module has been initiated. As discussed above, Python is a high-level, platform-independent, scripting language. The Python console allows the user to enter and execute Python commands and to save commands as a module. The Python command interpreter runs on the local agent, however, the syscalls generated by the command interpreter are executed by the syscall proxy server of the level 0 agent, which is the selected source.

In contrast, a module may run on a remote agent that has the capability to execute modules, such as a level 2 agent. As discussed above, the level 2 agent has a virtual machine that enables it to

ForcePoint LLC
EX-1009
Page277

execute scripting language modules. In such a case, the module is resident on and executes on the remote agent. Syscalls generated by the module are executed either as a native call on the remote host (see Figs. 7 and 8) or by a syscall proxy server on a downstream agent. The module may be installed together with the remote agent, or may be downloaded after the remote agent has been installed. This capability for 100% remote execution is advantageous, because it allows the module to run without receiving a continuous series of commands, thereby avoiding the generation of excessive network traffic. Moreover, a module for exploiting a security vulnerability may have certain timing requirements that, due to network latency, cannot be met if commands must be received over the network.

Regardless of whether a module is being run on the local agent or a remote agent, the syscalls generated by the module will be executed on the selected source, which in the example of Fig. 21 is the level 0 agent installed on target host 192.168.22.6. Thus, the Python commands executed in the Python console generate syscalls that are executed on this selected target host. For example, the Python command "print socket.gethostname()", shown in the Python console window, generates a syscall to retrieve the name of the target host, "diavolo". The remaining commands shown in the window generate syscalls to create a socket in the target host, which can be used to retrieve information from the target host.

It will be appreciated that each of these embodiments discussed above provides a novel system and method for analyzing computer system security by compromising the security in a systematic manner.

ForcePoint LLC
EX-1009
Page278

It also will be appreciated that because the system provides a database of modules for exploiting security vulnerabilities, the system described herein actually attempts to exploit detected vulnerabilities, rather than merely listing them.

It also will be appreciated that because the system provides a virtual machine to execute modules in agents installed in target hosts, the modules can be run on a variety of platforms and operating systems without further customization and testing. In addition, publicly available programs to exploit security vulnerabilities can be used without an extensive laboratory infrastructure for rewriting and testing.

It also will be appreciated that because the system maintains a database of all operations performed during the penetration testing, the target system can be reliably returned to its original configuration.

While the present invention has been described with respect to what is presently considered to be the preferred embodiments, it is to be understood that the invention is not limited to the disclosed embodiments. To the contrary, the invention is intended to cover various modifications and equivalent arrangements included within the spirit and scope of the appended claims.

ForcePoint LLC
EX-1009
Page279

WHAT IS CLAIMED:

1. A system for performing penetration testing of a target computer network by installing a remote agent in the target computer network, the system comprising:

a local agent provided in a console and configured to receive and execute commands;

a user interface provided in the console and configured to send commands to and receive information from the local agent, process the information, and present the processed information;

a database configured to store the information received from the local agent;

a network interface connected to the local agent and configured to communicate via a network with the remote agent installed in the target computer network; and

security vulnerability exploitation modules for execution by the local agent and/or the remote agent.

2. The system of claim 1, wherein the user interface enables a user to select one of the modules and initiate execution of the selected module on either the local agent or the remote agent.

3. The system of claim 1, wherein the user interface provides a graphical representation of the target computer network.

4. A method for performing penetration testing of a target computer network, comprising:

ForcePoint LLC
EX-1009
Page280

installing a remote agent in the target computer network;

executing a command using a local agent provided in a console;

receiving information from the local agent in a user interface provided in the console;

presenting the information received from the local agent to a user;

storing the information received from the local agent in a database;

communicating via a network with the remote agent installed in the target computer network; and

providing security vulnerability exploitation modules for execution by the local agent and/or the remote agent.

5. The method of claim 4, further comprising:

selecting, using the user interface, one of the modules; and

initiating execution of the selected module on either the local agent or the remote agent.

6. The method of claim 4, further comprising providing a graphical representation of the target computer network using the user interface.

7. Computer code for performing penetration testing of a target computer network, the computer code comprising code for:

installing a remote agent in the target computer network;

executing a command using a local agent provided in a console;

receiving information from the local agent in a user interface provided in the console;

presenting the information received from the local agent to a user;

storing the information received from the local agent in a database;

communicating via a network with the remote agent installed in the target computer network; and

providing security vulnerability exploitation modules for execution by the local agent and/or the remote agent.

8. The computer code of claim 7, further comprising code for:

selecting, using the user interface, one of the modules; and

initiating execution of the selected module on either the local agent or the remote agent.

9. The computer code of claim 7, further comprising code for providing a graphical representation of the target computer network using the user interface.

10. An agent for use in a system for performing penetration testing of a target computer network, the agent comprising:

a proxy server configured to receive and execute system calls received via a network; and

a virtual machine configured to execute scripting language instructions received via the network.

ForcePoint LLC
EX-1009
Page282

11. The agent of claim 10, further comprising an execution engine configured to control the proxy server and the virtual machine, wherein the system calls and the scripting language instructions are routed to the proxy server and the virtual machine, respectively, by the execution engine.

12. The agent of claim 11, further comprising a remote procedure call module configured to receive commands from the network formatted in a remote procedure call protocol and pass the commands to the execution engine.

13. An agent for use in a system for performing penetration testing of a target computer network, the agent comprising:

a proxy server configured to receive and execute system calls received via a network;

a virtual machine configured to execute scripting language instructions received via the network;

a secure communication module configured to provide secure communication between the virtual machine and the network;

an execution engine configured to control the proxy server and the virtual machine, wherein the system calls and the scripting language instructions are routed to the proxy server and the virtual machine, respectively, by the execution engine;

a remote procedure call module configured to receive commands via the network formatted in a remote procedure call protocol and pass the commands to the execution engine; and

ForcePoint LLC
EX-1009
Page283

a second secure communication module configured to provide secure communication between the remote procedure call module and the network.

14. A method for performing penetration testing of a target network, comprising the steps of:

executing a first module in a console having a user interface, the first module being configured to exploit a security vulnerability in a first target host of the target network;

installing a first remote agent in the first target host, the first remote agent being configured to communicate with the console and a second remote agent; and

executing a second module in the first remote agent, the second module being configured to exploit a security vulnerability in a second target host of the target network.

15. The method of claim 14, further comprising installing a second remote agent in the second target host of the target network, the second remote agent being configured to communicate with the first remote agent.

16. A system for performing penetration testing of a target network, comprising:

a console having a user interface;

a first module configured to execute in the console to exploit a security vulnerability in a first target host of the target network;

a first remote agent installed in the first target host, the first remote agent being configured to communicate with the console and a second remote agent; and

ForcePoint LLC
EX-1009
Page284

a second module configured to execute in the first remote agent to exploit a security vulnerability in a second target host of the target network.

17. The system of claim 16, further comprising a second remote agent installed in the second target host of the target network, the second remote agent being configured to communicate with the first remote agent.

18. Computer code for performing penetration testing of a target network, the computer code comprising code for:

executing a first module in a console having a user interface, the first module being configured to exploit a security vulnerability in a first target host of the target network;

installing a first remote agent in the first target host, the first remote agent being configured to communicate with the console and a second remote agent; and

executing a second module in the first remote agent, the second module being configured to exploit a security vulnerability in a second target host of the target network.

19. The computer code of claim 18, further comprising code for installing a second remote agent in the second target host of the target network, the second remote agent being configured to communicate with the first remote agent.

20. A method for performing penetration testing of a target network, comprising the steps of:

ForcePoint LLC
EX-1009
Page285

executing a first module to exploit a security vulnerability of a first target host of the target network;

installing a first remote agent in the first target host as a result of exploiting the security vulnerability of the first target host;

sending a system call to the first remote agent via a network; and

executing the system call in the first target host using a proxycall server of the first remote agent to exploit a security vulnerability of a second target host.

21. A method for performing penetration testing of a target network, comprising the steps of:

executing a first module to exploit a security vulnerability of a first target host of the target network;

installing a first remote agent in the first target host as a result of exploiting the security vulnerability of the first target host;

executing in the first remote agent a second module that generates a system call; and

executing the system call in the first target host to exploit a security vulnerability of a second target host.

22. A method for performing penetration testing of a target network, comprising the steps of:

executing a first module to exploit a security vulnerability of a first target host of the target network;

ForcePoint LLC
EX-1009
Page286

installing a first remote agent in the first target host as a result of exploiting the security vulnerability of the first target host;

executing a second module in the first remote agent that generates a system call;

installing a second remote agent in the second target host as a result of exploiting a security vulnerability of the second target host;

sending the system call generated by the second module to the second remote agent via a network; and

executing the system call in the second target host using a proxycall server of the second remote agent.

23. A method for performing penetration testing of a target network, comprising the steps of:

executing a first module to exploit a security vulnerability of a first target host of the target network;

installing a first remote agent in the first target host as a result of exploiting the security vulnerability of the first target host;

installing a second remote agent in the second target host as a result of exploiting a security vulnerability of the second target host;

sending a system call to the first remote agent;

sending the system call from the first remote agent to the second remote agent; and

executing the system call in the second target host using a proxycall server of the second remote agent.

ForcePoint LLC
EX-1009
Page287

24. A method for performing penetration testing of a target network, comprising the steps of:

installing a first remote agent in the first target host, the first remote agent having a proxy server configured to receive and execute system calls;

executing in the first remote agent a system call received via a network;

installing a second remote agent in the first target host, the second remote agent having a proxy server configured to receive and execute system calls and a virtual machine configured to execute scripting language instructions; and

executing in the second remote agent a scripting language instruction or a system call, the system call being received via the network.

25. Computer code for performing penetration testing of a target network, the code comprising code for:

executing a first module to exploit a security vulnerability of a first target host of the target network;

installing a first remote agent in the first target host as a result of exploiting the security vulnerability of the first target host;

sending a system call to the first remote agent via a network; and

executing the system call in the first target host using a proxycall server of the first remote agent to exploit a security vulnerability of a second target host.

26. Computer code for performing penetration testing of a target network, the code comprising code for:

ForcePoint LLC
EX-1009
Page288

executing a first module to exploit a security vulnerability of a first target host of the target network;

installing a first remote agent in the first target host as a result of exploiting the security vulnerability of the first target host;

executing in the first remote agent a second module that generates a system call; and

executing the system call in the first target host to exploit a security vulnerability of a second target host.

27. Computer code for performing penetration testing of a target network, the code comprising code for:

executing a first module to exploit a security vulnerability of a first target host of the target network;

installing a first remote agent in the first target host as a result of exploiting the security vulnerability of the first target host;

executing a second module in the first remote agent that generates a system call;

installing a second remote agent in the second target host as a result of exploiting a security vulnerability of the second target host;

sending the system call generated by the second module to the second remote agent via a network; and

executing the system call in the second target host using a proxycall server of the second remote agent.

ForcePoint LLC
EX-1009
Page289

28. Computer code for performing penetration testing of a target network, the code comprising code for:

executing a first module to exploit a security vulnerability of a first target host of the target network;

installing a first remote agent in the first target host as a result of exploiting the security vulnerability of the first target host;

installing a second remote agent in the second target host as a result of exploiting a security vulnerability of the second target host;

sending a system call to the first remote agent;

sending the system call from the first remote agent to the second remote agent; and

executing the system call in the second target host using a proxycall server of the second remote agent.

29. Computer code for performing penetration testing of a target network, the code comprising code for:

installing a first remote agent in the first target host, the first remote agent having a proxy server configured to receive and execute system calls;

executing in the first remote agent a system call received via a network;

installing a second remote agent in the first target host, the second remote agent having a proxy server configured to receive and execute system calls and a virtual machine configured to execute scripting language instructions; and

executing in the second remote agent a scripting language instruction or a system call, the system call being received via the network.

ForcePoint LLC
EX-1009
Page290

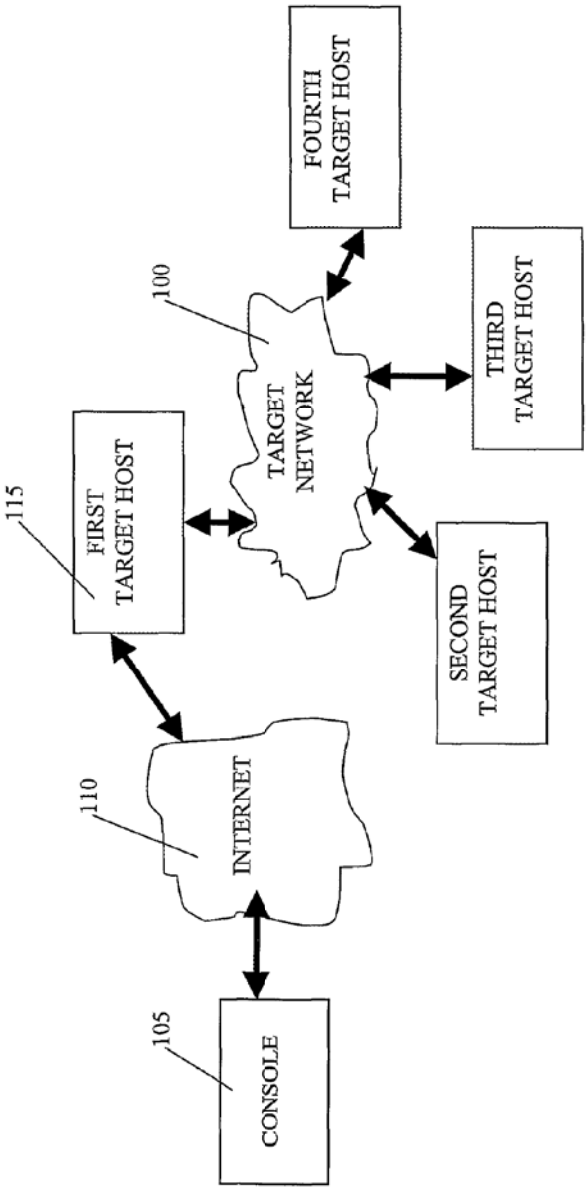


Fig. 1

ForcePoint LLC
EX-1009
Page291

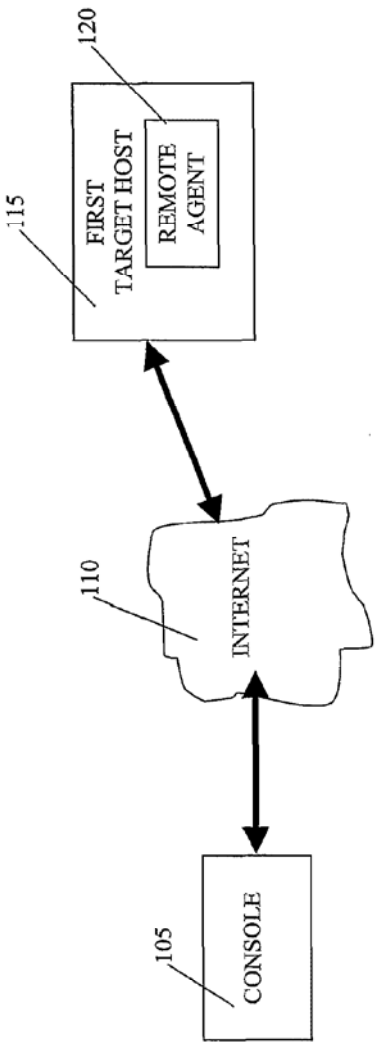


Fig. 2

ForcePoint LLC
EX-1009
Page292

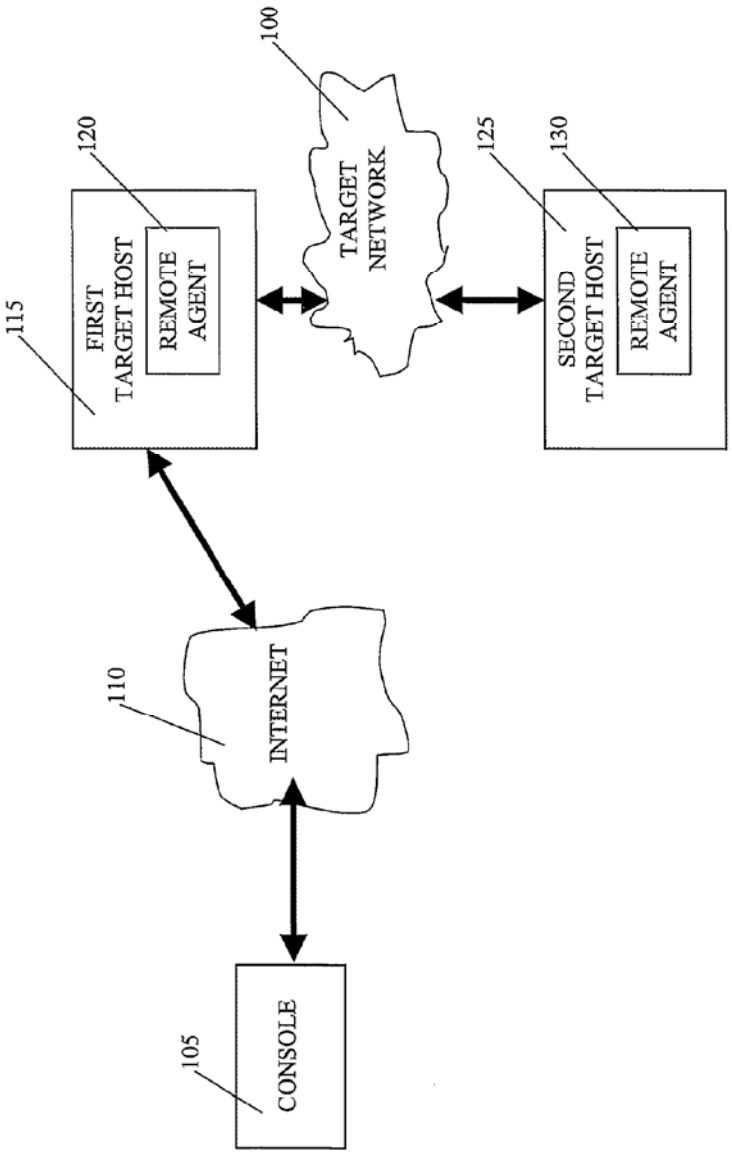

100
TARGET NETWORK
120
REMOTE AGENT
FIRST TARGET HOST
115
125
130
SECOND TARGET HOST
REMOTE AGENT
INTERNET
110
CONSOLE
105

Fig. 3

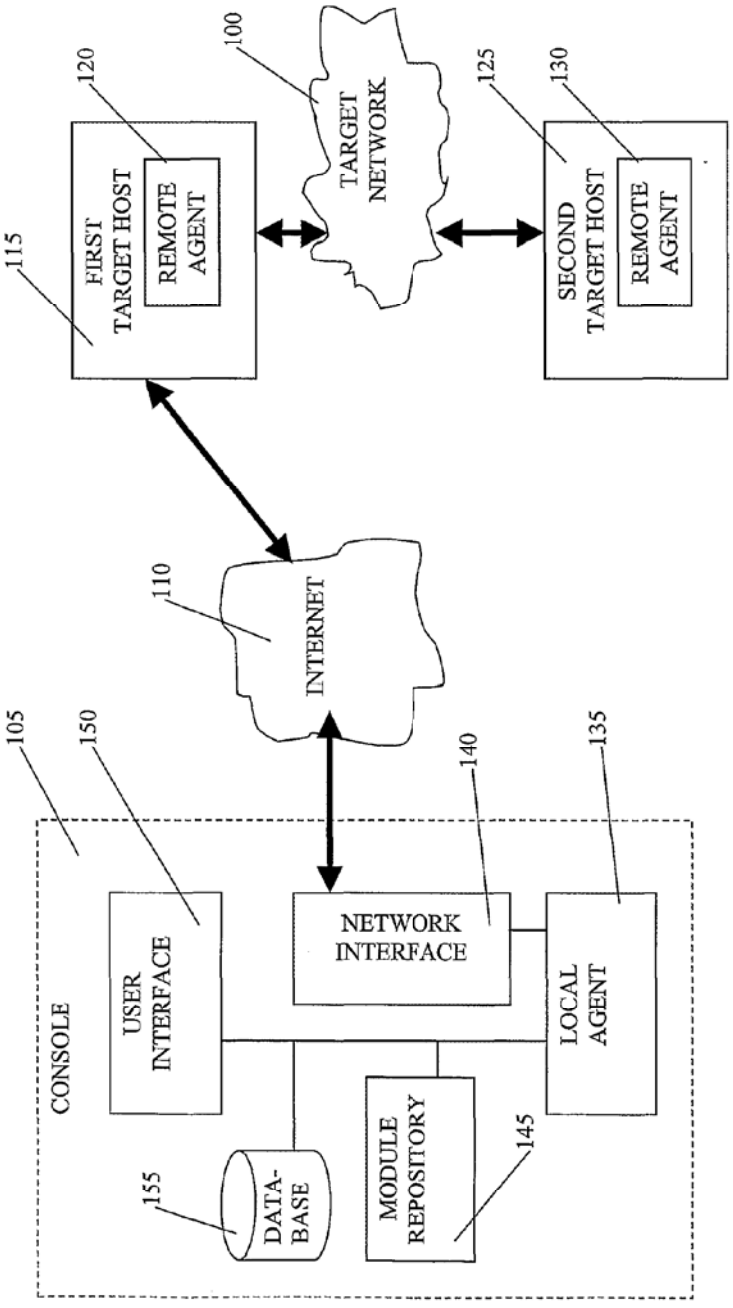


Fig. 4

ForcePoint LLC
EX-1009
Page294

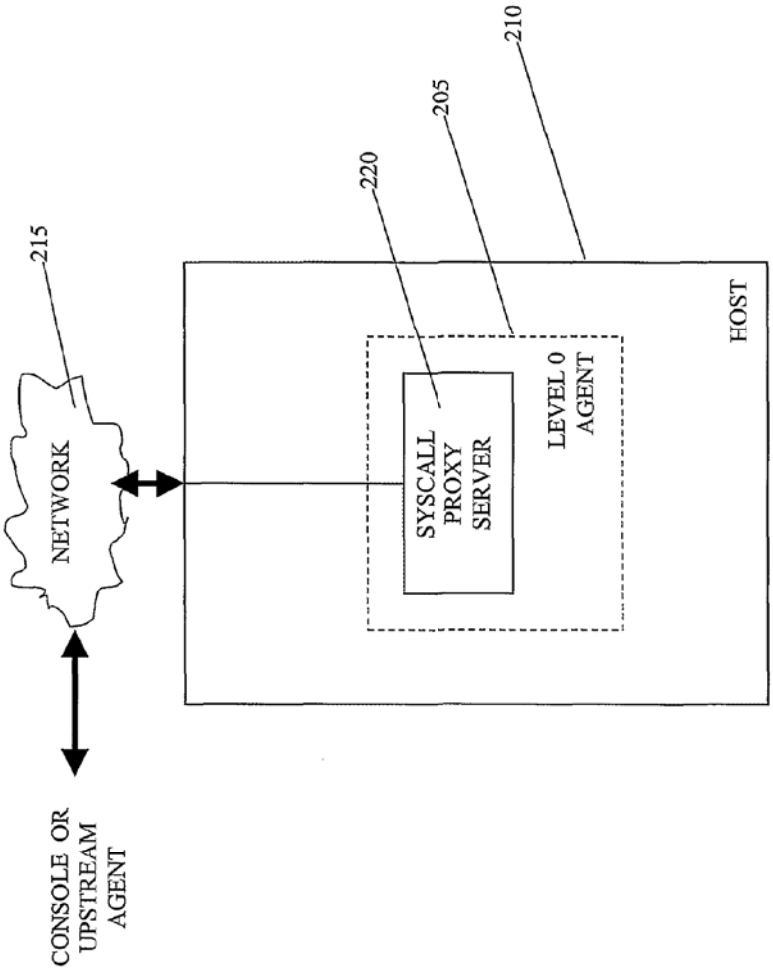

210
205
220
215
HOST
LEVEL 0 AGENT
SYSCALL PROXY SERVER
NETWORK
CONSOLE OR UPSTREAM AGENT

Fig. 5

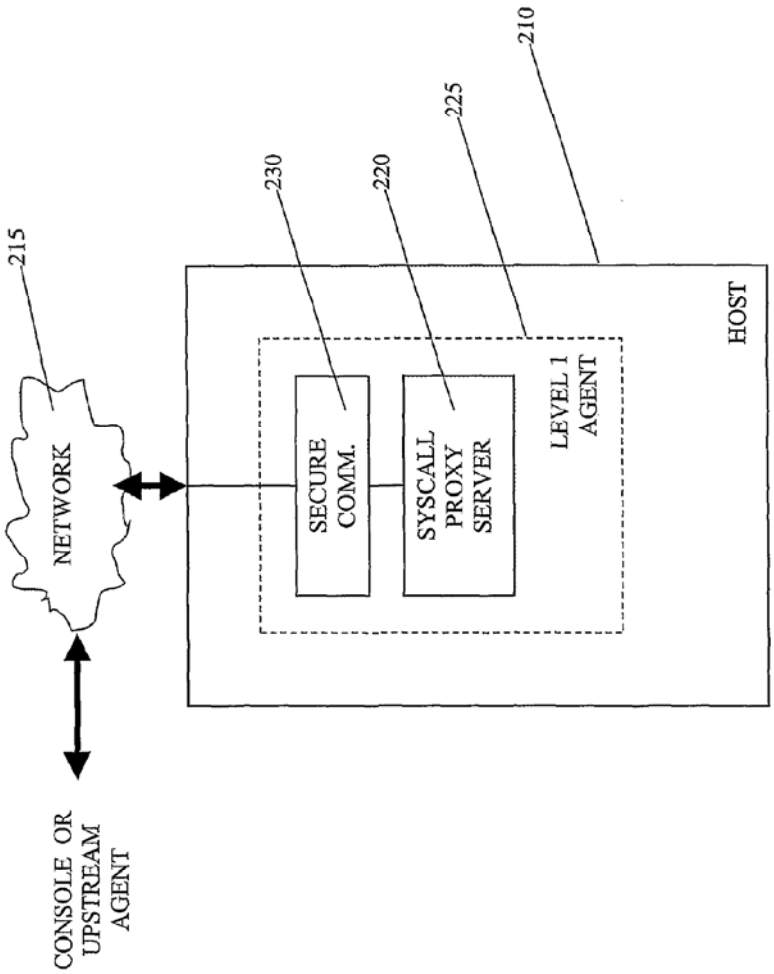

210
225
220
230
215
HOST
LEVEL 1 AGENT
SECURE COMM.
SYSCALL PROXY SERVER
NETWORK
CONSOLE OR UPSTREAM AGENT

Fig. 6

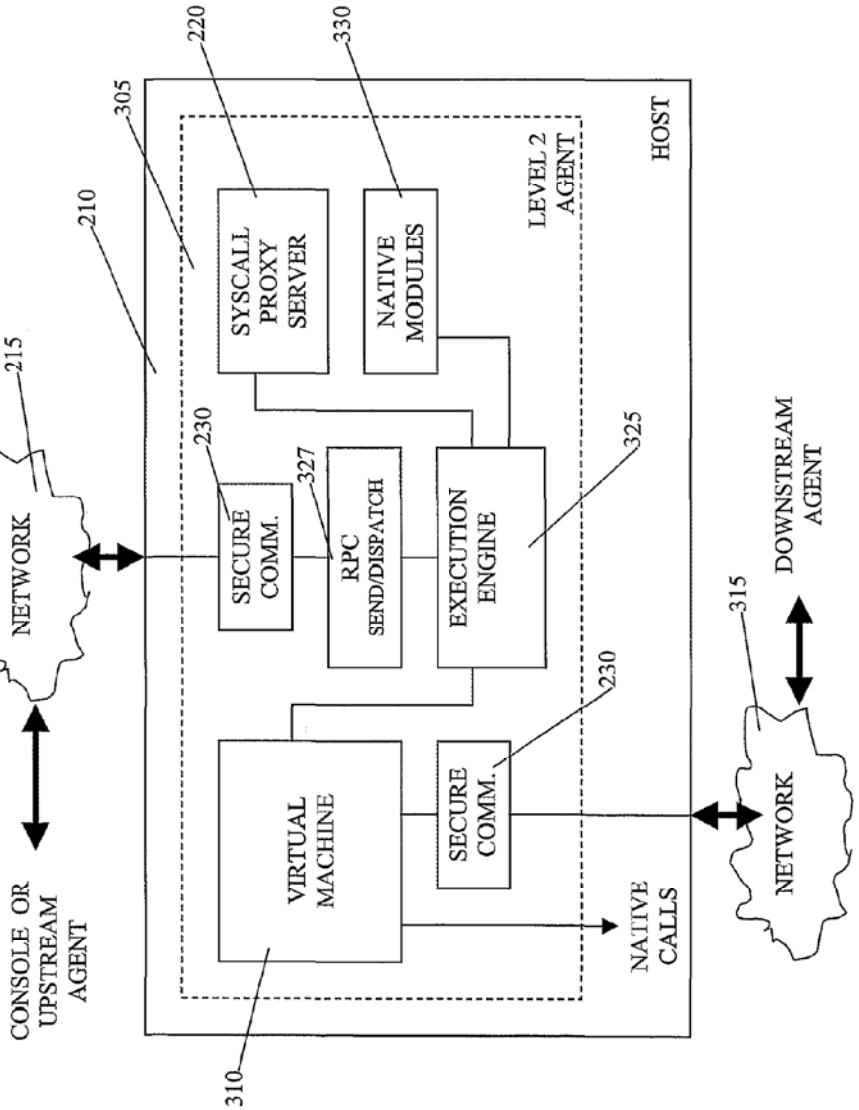


Fig. 7

ForcePoint LLC
EX-1009
Page297

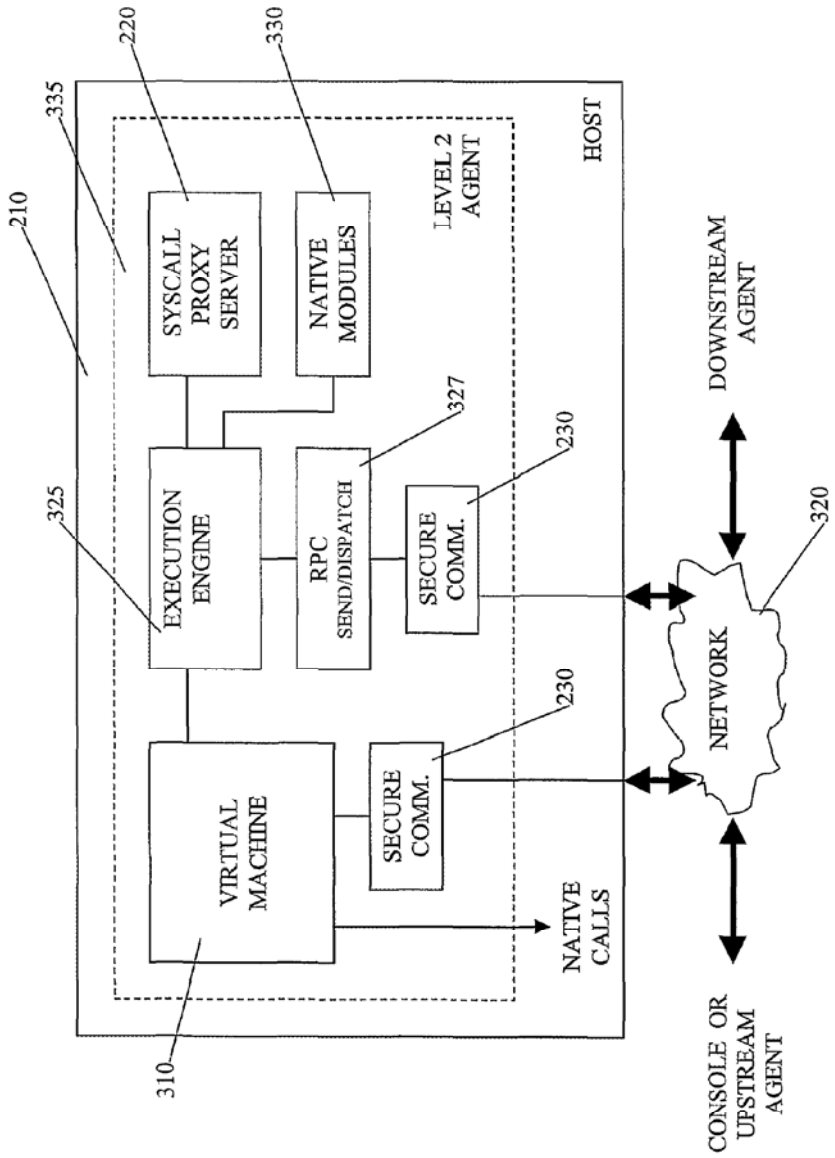


Fig. 8

ForcePoint LLC
EX-1009
Page298

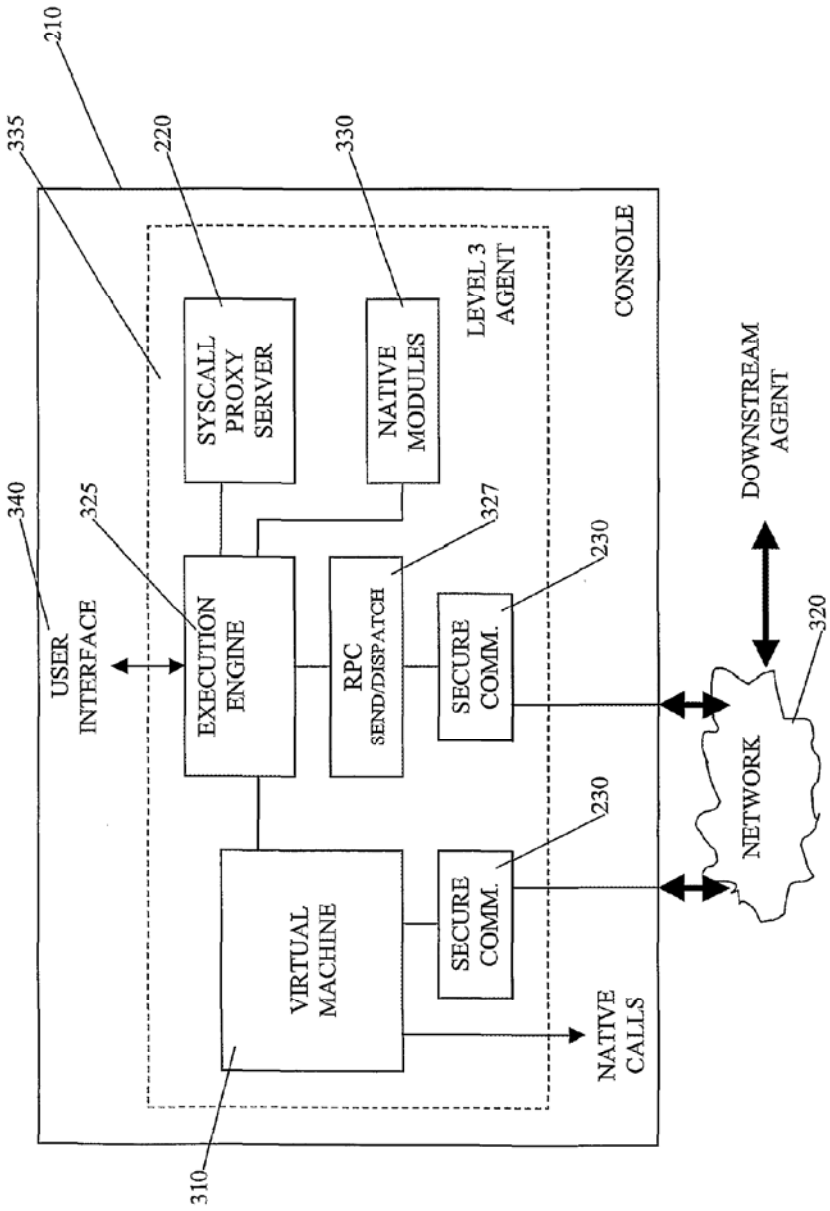

210
335
220
330
CONSOLE
LEVEL 3 AGENT
SYSCALL PROXY SERVER
NATIVE MODULES
DOWNSTREAM AGENT
340
325
327
230
USER INTERFACE
EXECUTION ENGINE
RPC SEND/DISPATCH
SECURE COMM.
320
NETWORK
230
VIRTUAL MACHINE
SECURE COMM.
NATIVE CALLS
310

Fig. 9

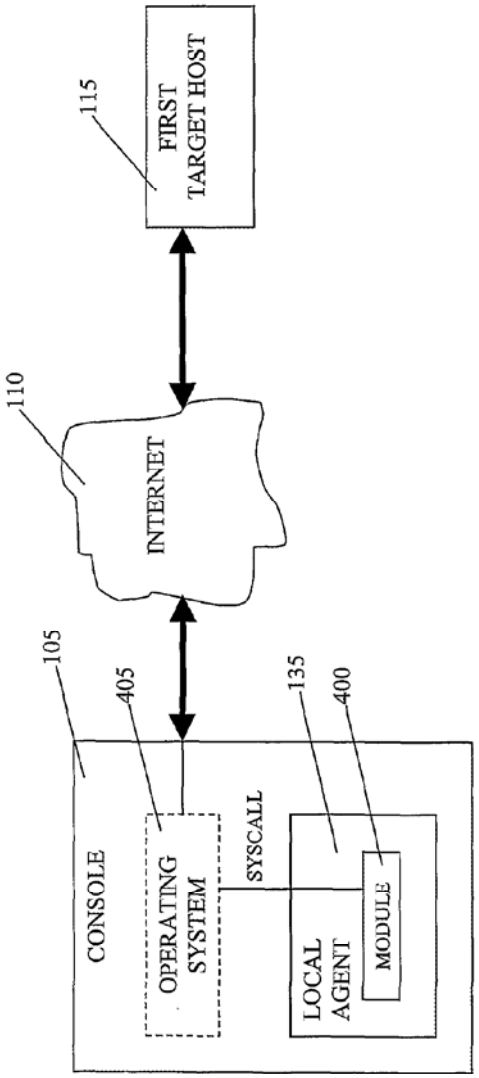

115
FIRST TARGET HOST
110
INTERNET
105
405
135
400
CONSOLE
OPERATING SYSTEM
SYSCALL
LOCAL AGENT
MODULE

Fig. 10

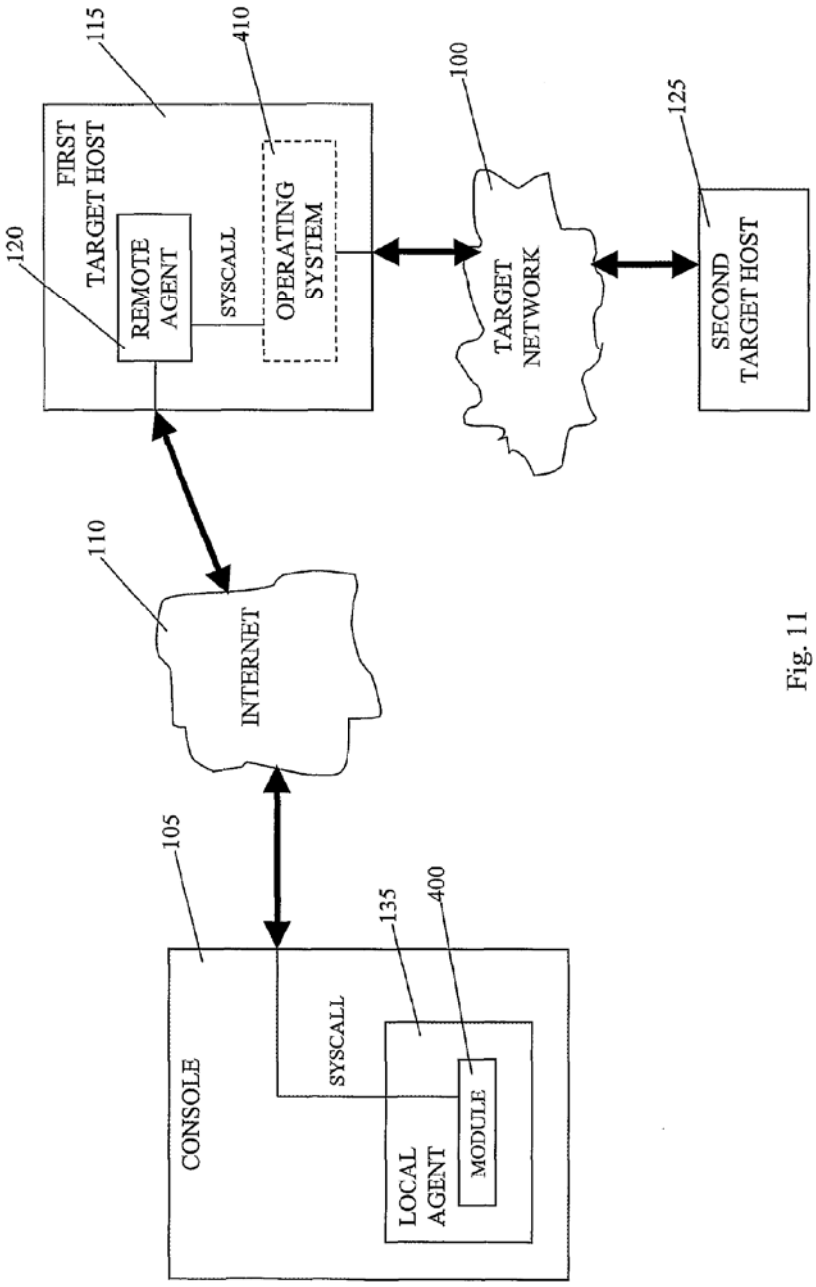

115
410
100
125
FIRST TARGET HOST
REMOTE AGENT
SYSCALL
OPERATING SYSTEM
120
TARGET NETWORK
SECOND TARGET HOST
110
INTERNET
105
135
400
CONSOLE
SYSCALL
LOCAL AGENT
MODULE

Fig. 11

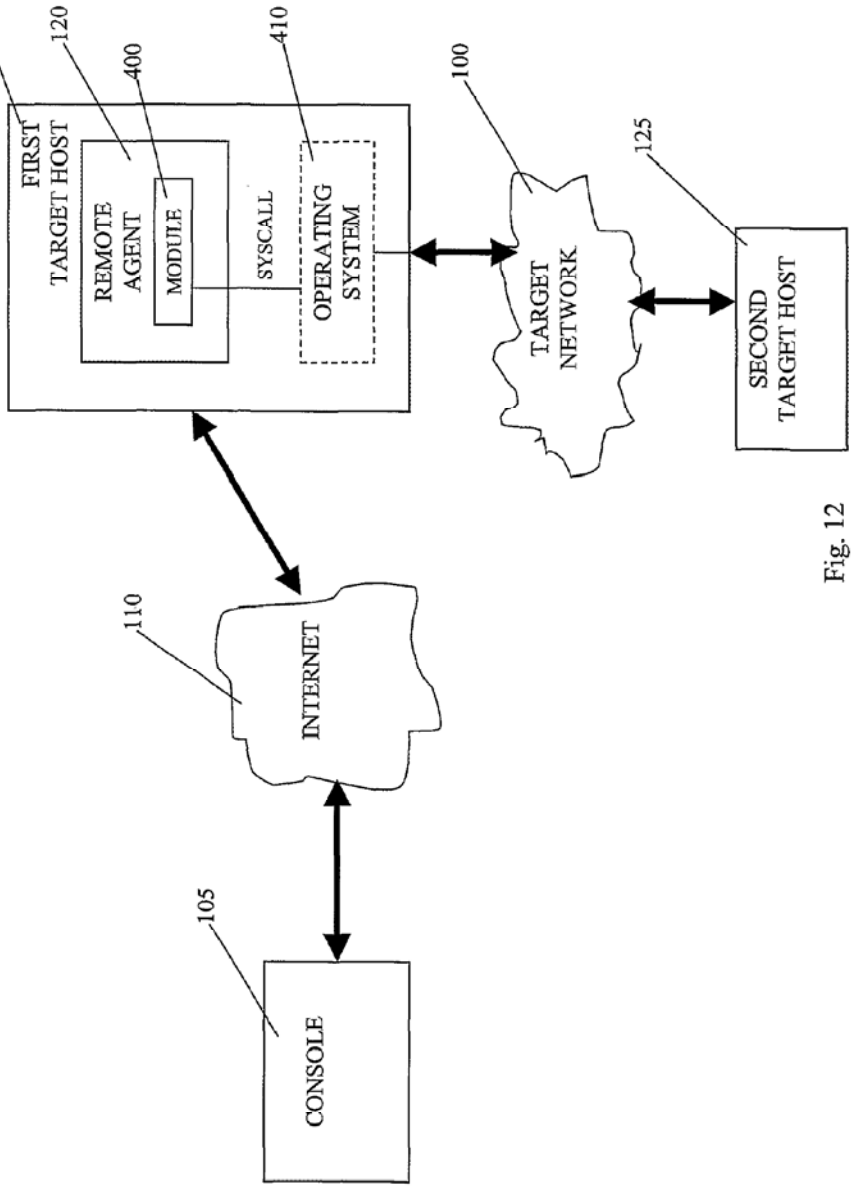

115
120
400
410
100
125
FIRST TARGET HOST
REMOTE AGENT
MODULE
SYSCALL
OPERATING SYSTEM
TARGET NETWORK
SECOND TARGET HOST
110
INTERNET
105
CONSOLE

Fig. 12

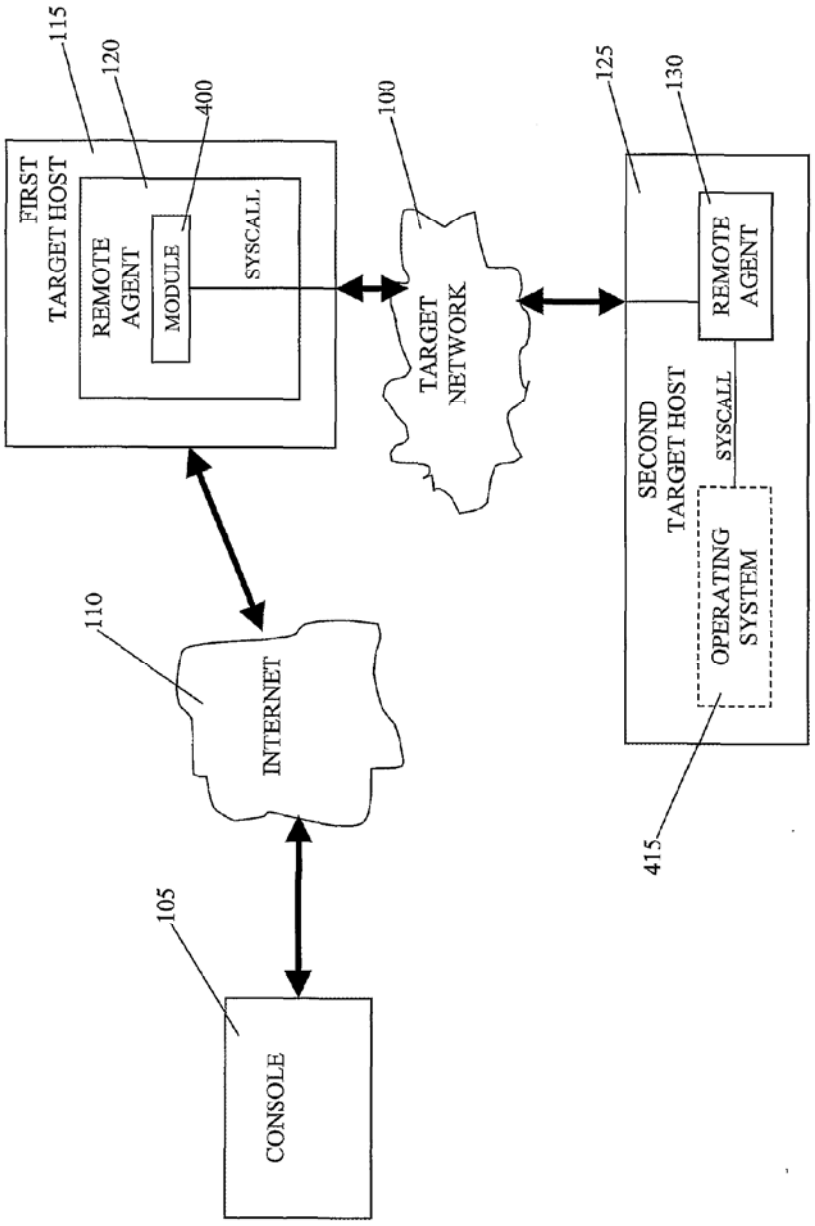

115
120
400
100
125
130
FIRST TARGET HOST
REMOTE AGENT
MODULE
SYSCALL
TARGET NETWORK
SECOND TARGET HOST
REMOTE AGENT
SYSCALL
OPERATING SYSTEM
110
INTERNET
105
CONSOLE
415

Fig. 13

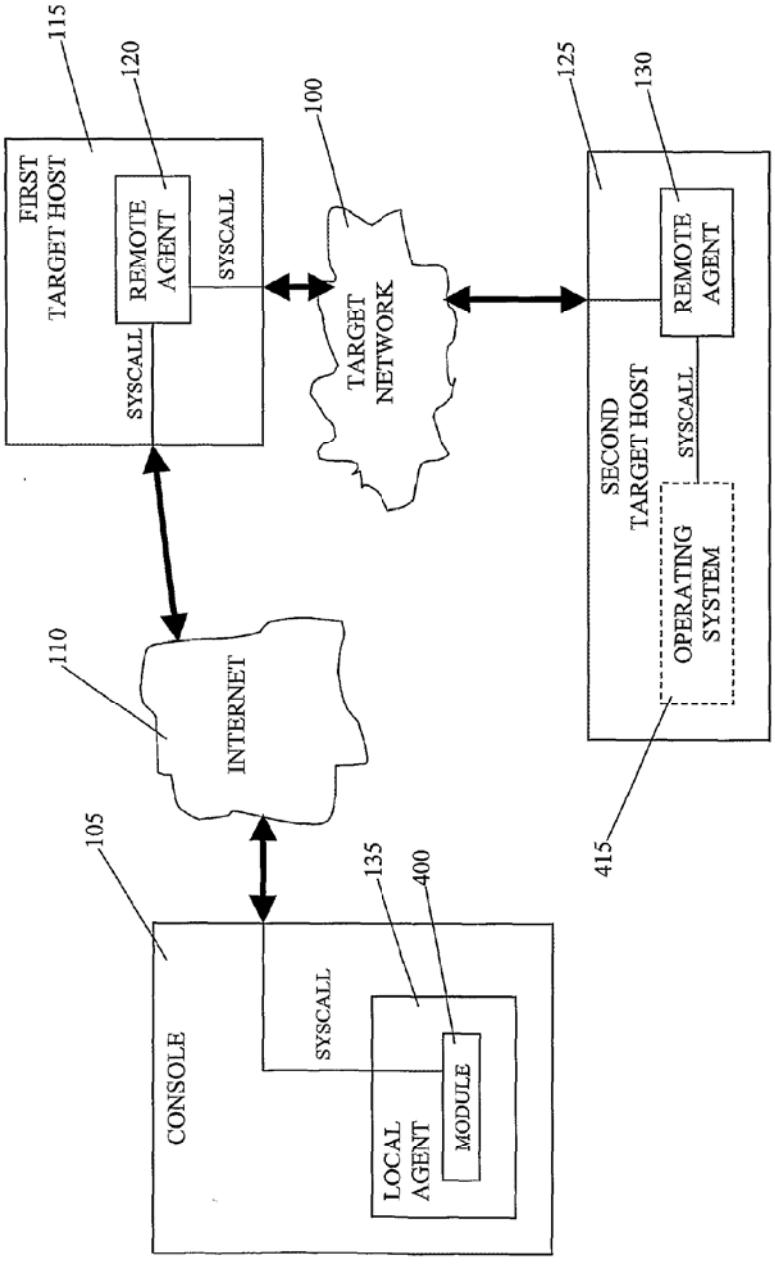

115
120
100
125
130
FIRST TARGET HOST
REMOTE AGENT
SYSCALL
SYSCALL
TARGET NETWORK
REMOTE AGENT
SECOND TARGET HOST
SYSCALL
OPERATING SYSTEM
110
INTERNET
105
135
400
415
CONSOLE
SYSCALL
LOCAL AGENT
MODULE

Fig. 14

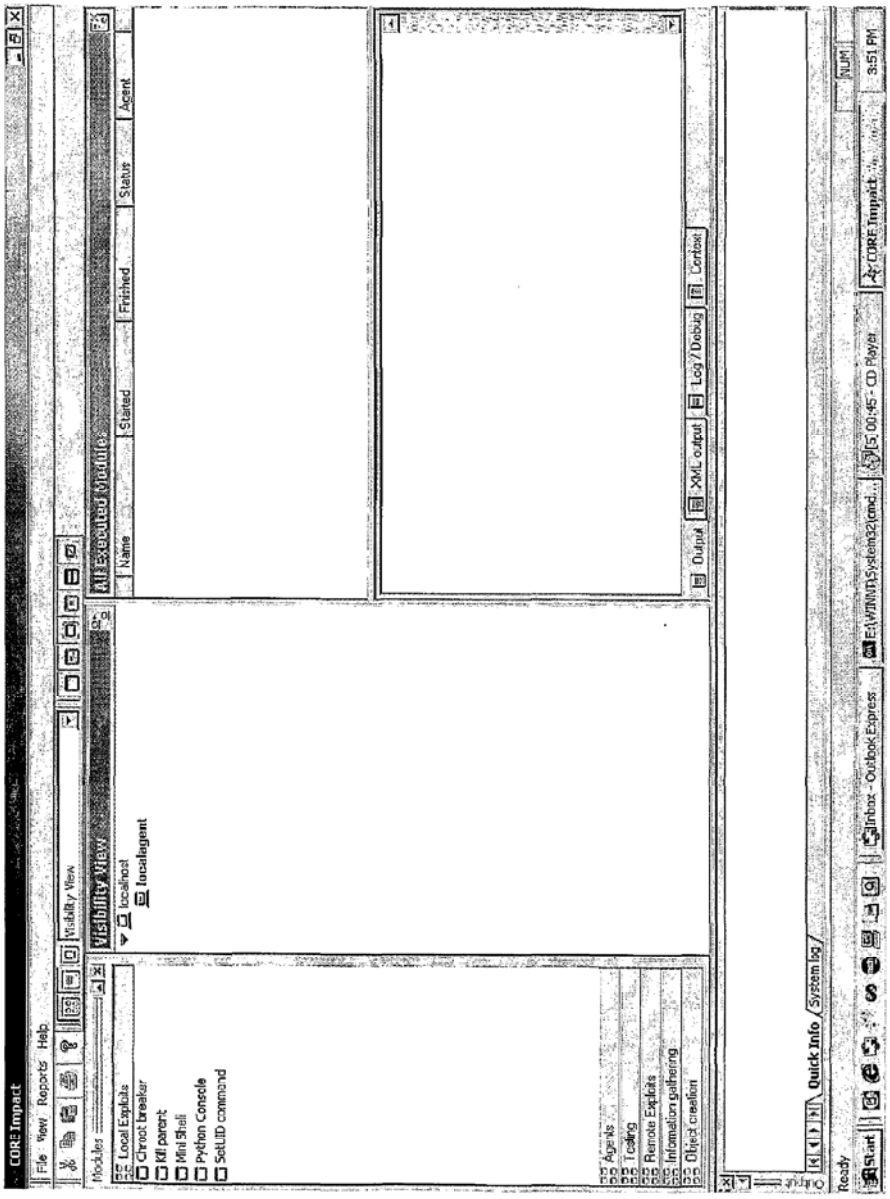


Fig. 15

ForcePoint LLC
EX-1009
Page305

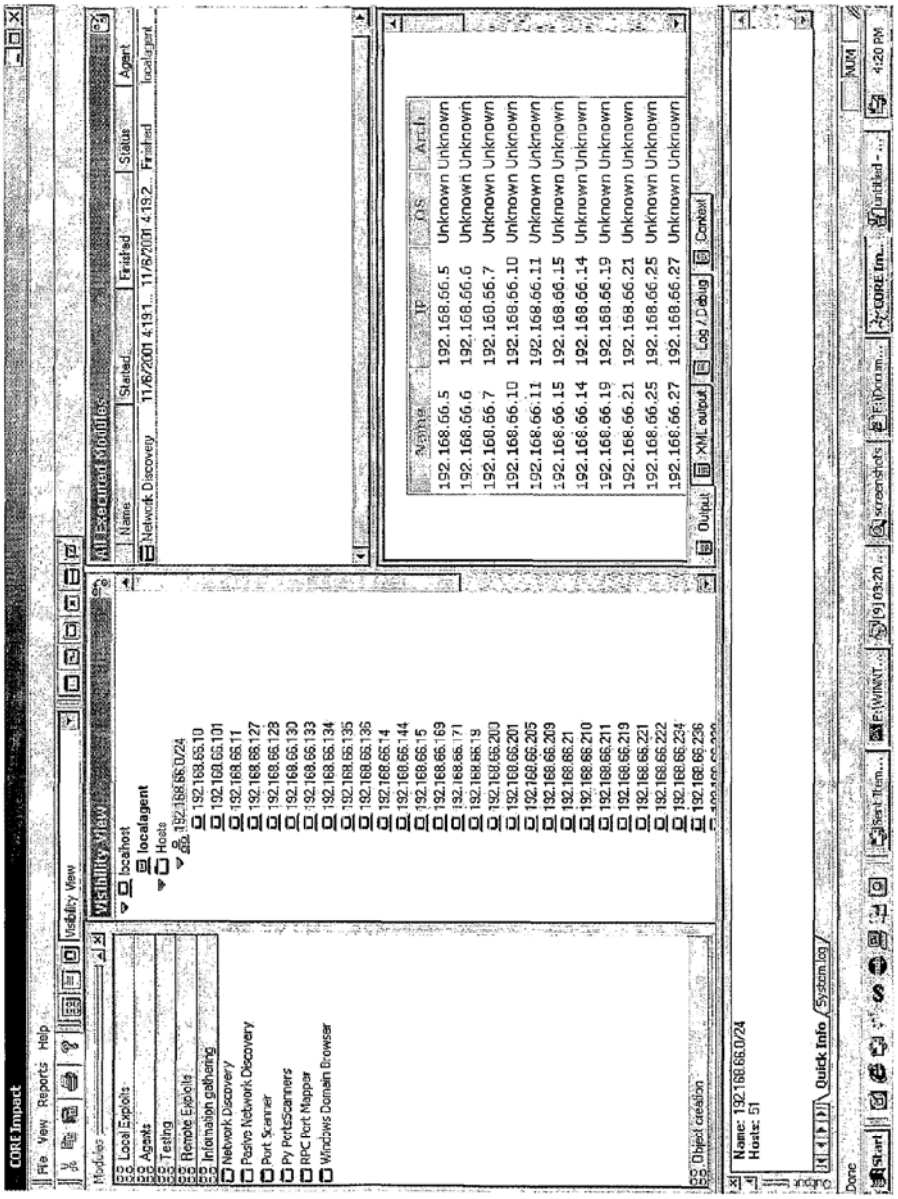


Fig. 16

ForcePoint LLC
EX-1009
Page306

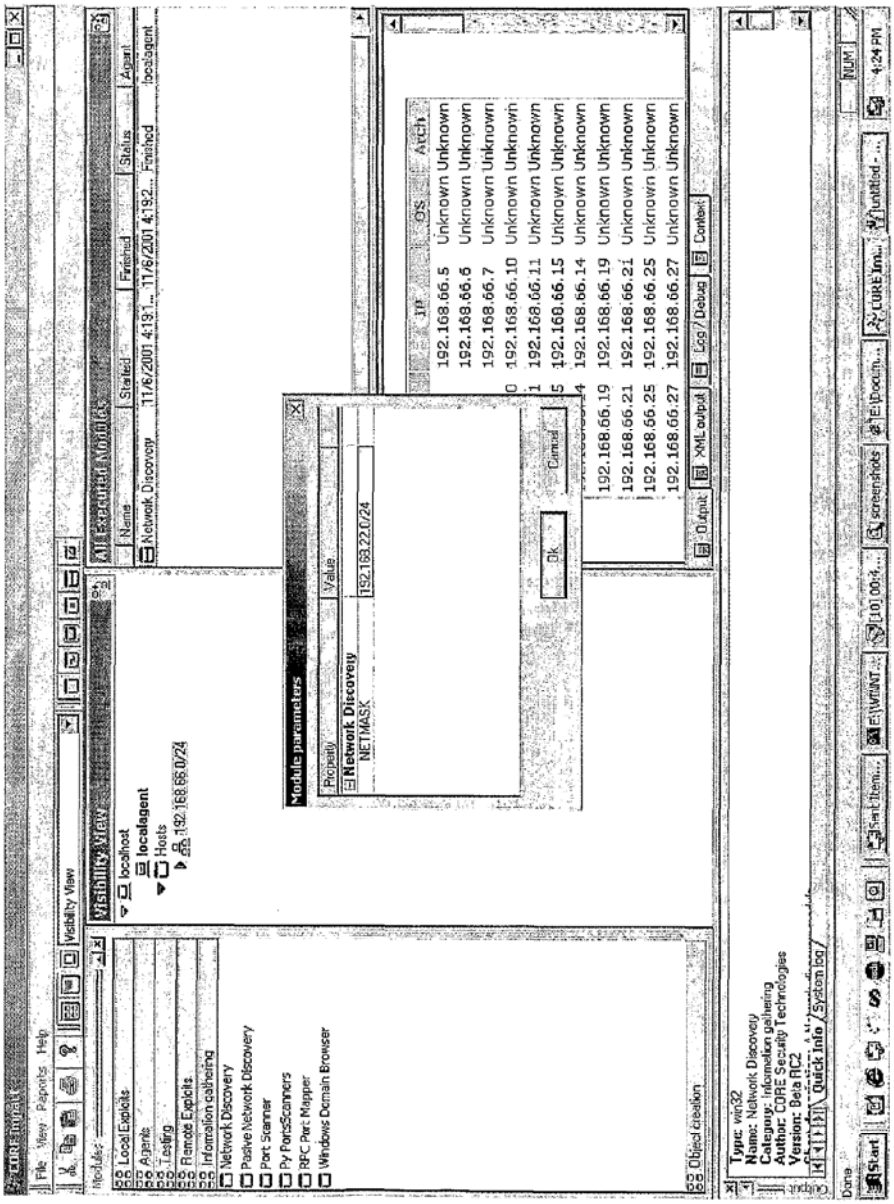


Fig. 17

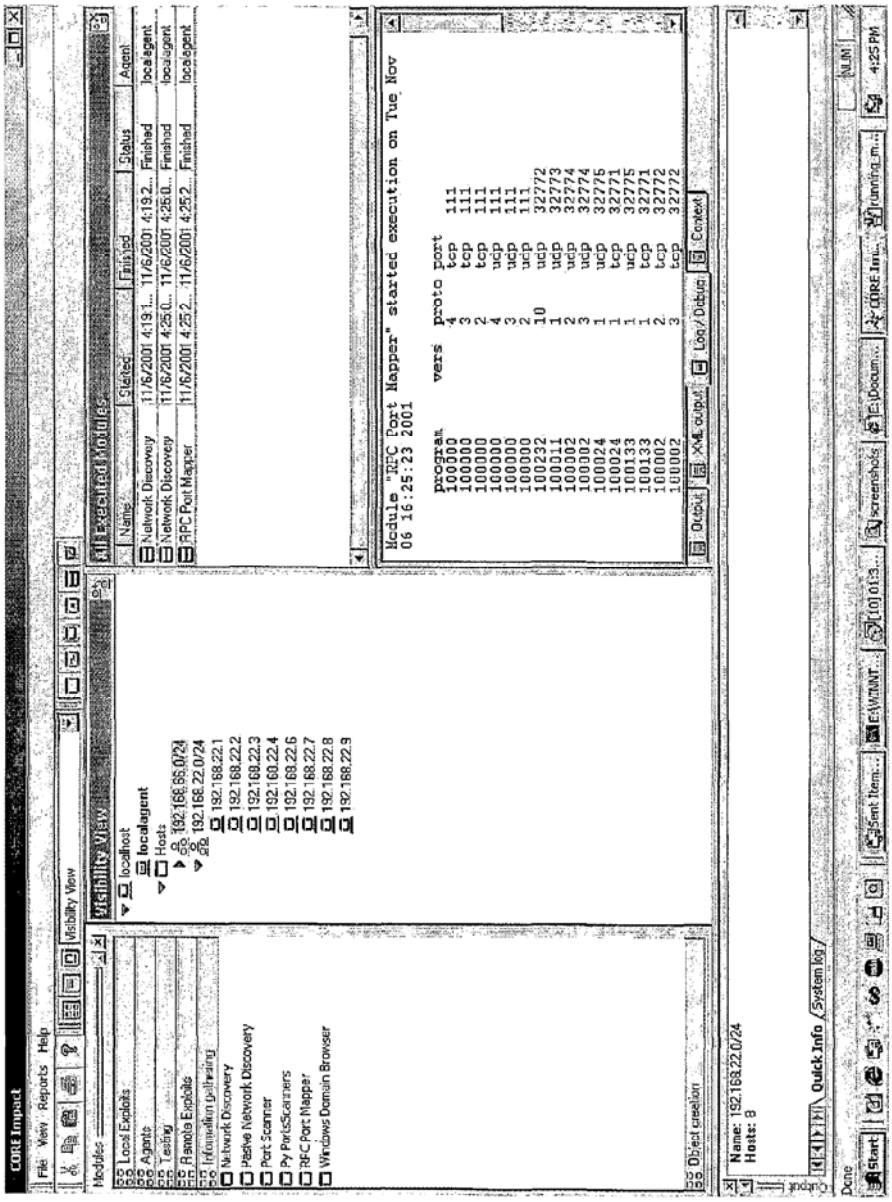


Fig. 18

ForcePoint LLC
EX-1009
Page308

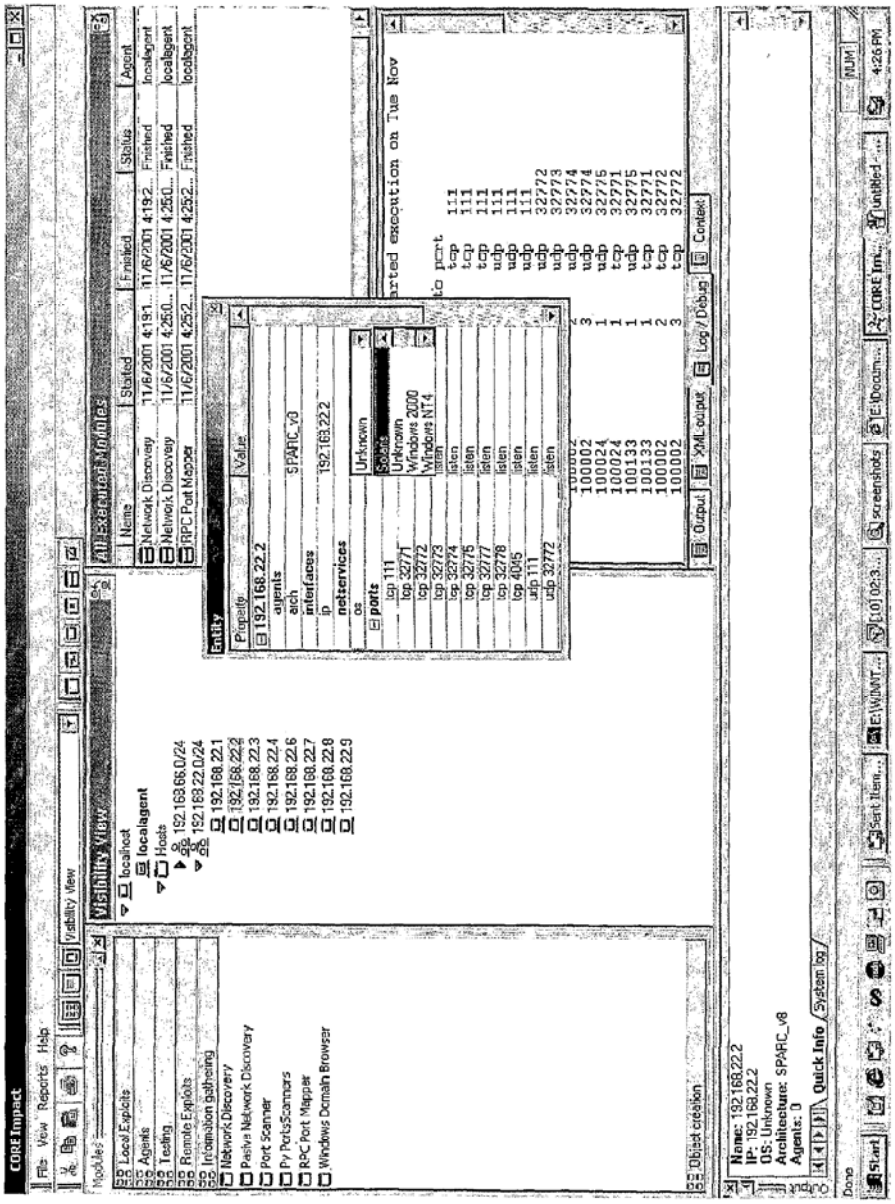


Fig. 19

ForcePoint LLC
EX-1009
Page309

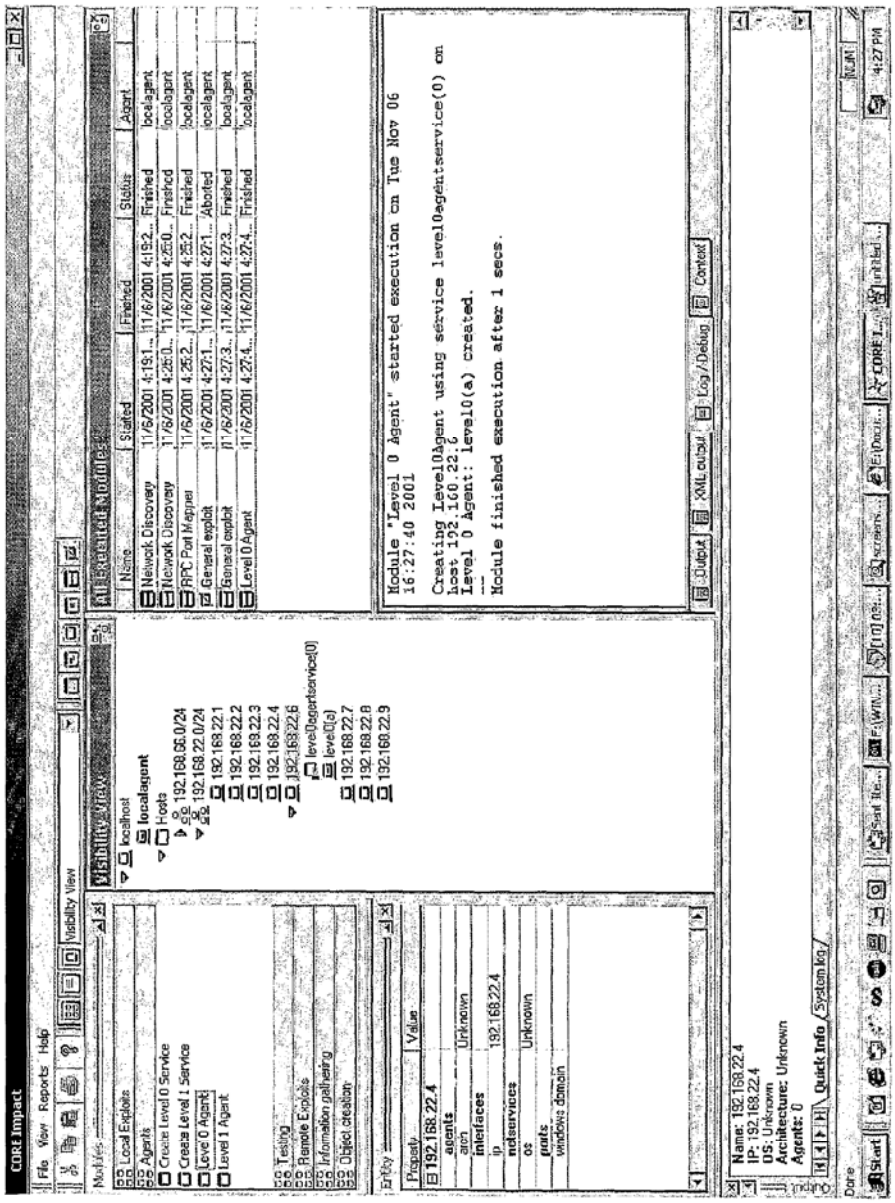


Fig. 20

ForcePoint LLC
EX-1009
Page310

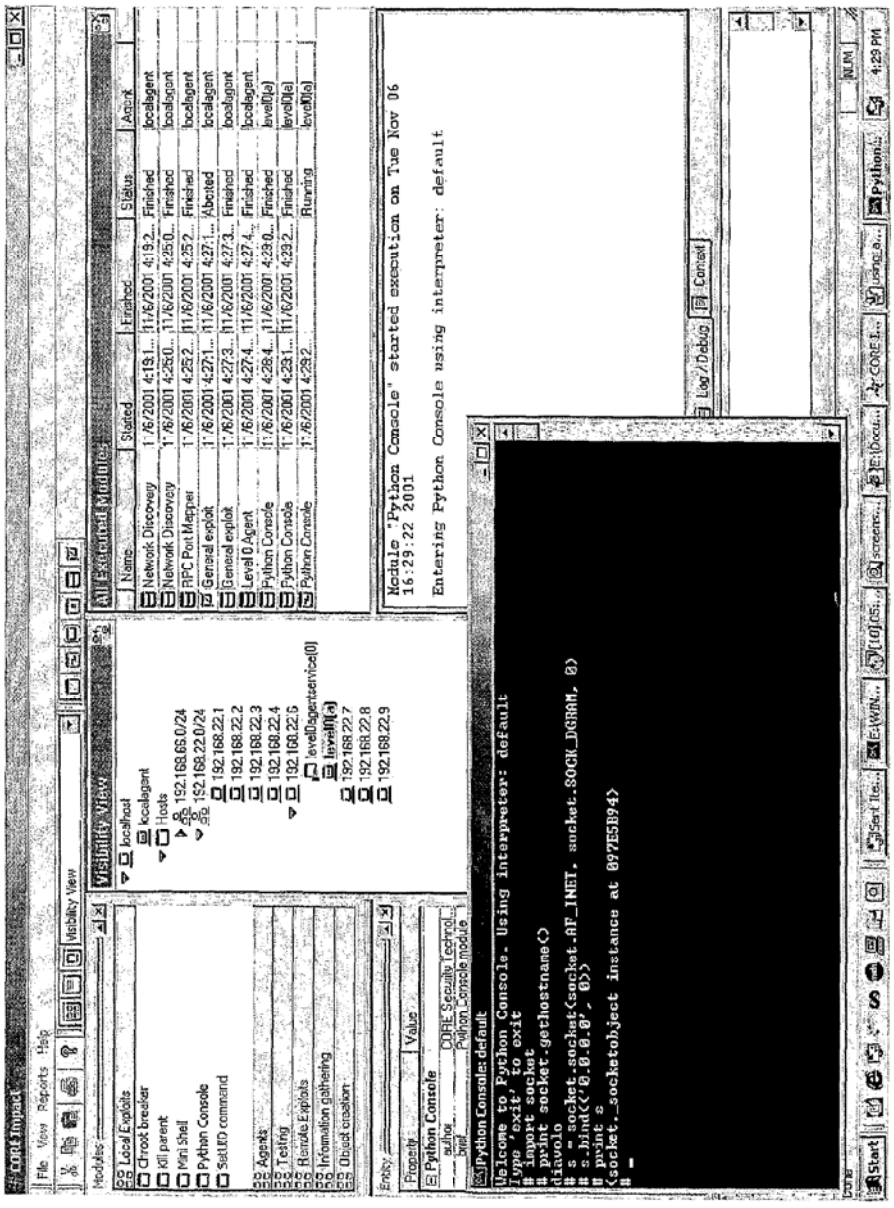


Fig. 21

ForcePoint LLC
EX-1009
Page311

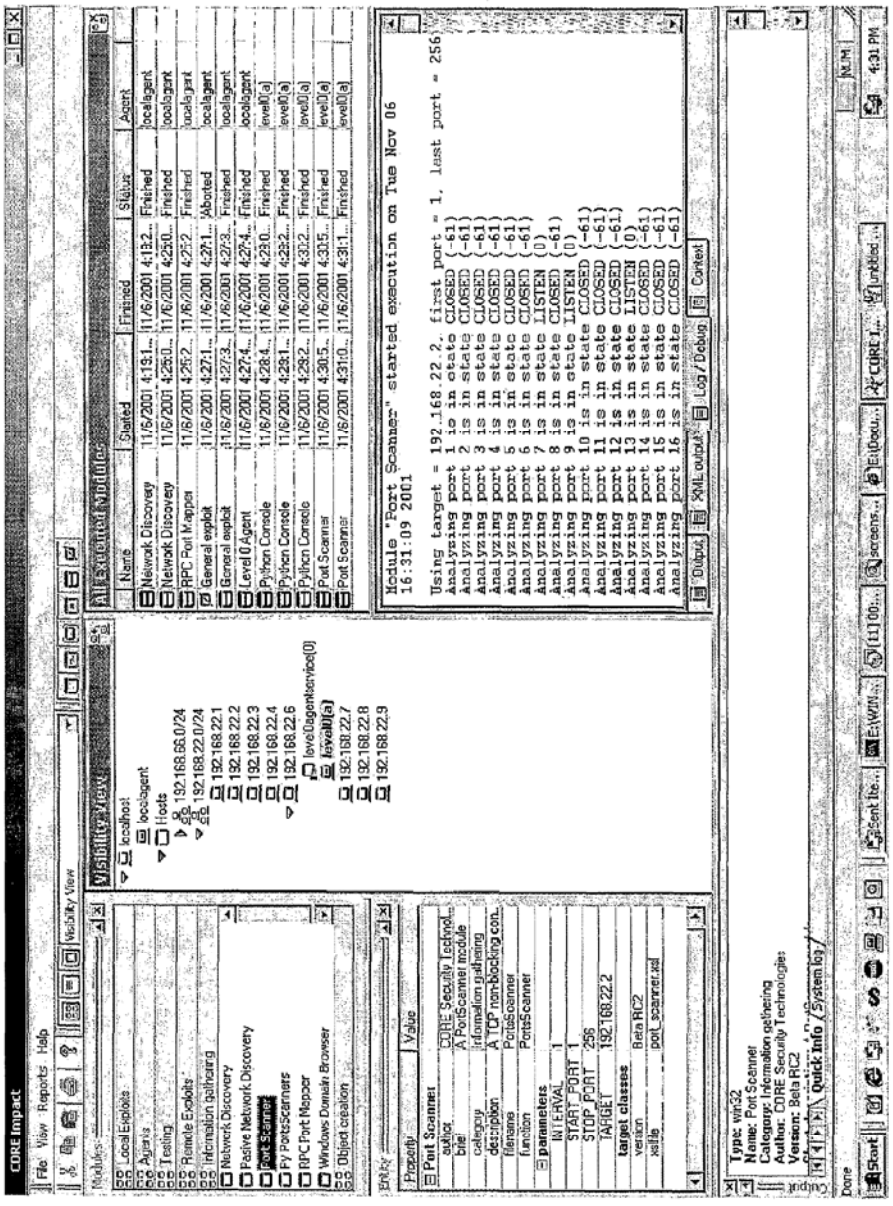


Fig. 22

ForcePoint LLC
EX-1009
Page312

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US02/22457

A. CLASSIFICATION OF SUBJECT MATTER
IPC(7) : G06F 17/30
US CL : 707/104.1, 707/10
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 707/104.1, 707/10

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

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 6,070,244 A (ORCHIER ET AL) 30 May 2000 (30.05.2000), column 1, lines 50-67; column 2, lines 1-67	1-29
A,P	US 6,408,391 B1 (HUFF ET AL) 18 June 2002 (18.06.2002), column 2, lines 55-67; column 3, lines 1-67	1-29

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
12 September 2002 (12.09.2002)	30 OCT 2002
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703)305-3230	Authorized officer Apu Mofiz Telephone No. 7036054240

Form PCT/ISA/210 (second sheet) (July 1998)

ForcePoint LLC
EX-1009
Page313

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 April 2003 (10.04.2003)

PCT

(10) International Publication Number
WO 03/029940 A2

(51) International Patent Classification[7]: G06F 1/00

(21) International Application Number: PCT/US02/26092

(22) International Filing Date: 15 August 2002 (15.08.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/969,686 2 October 2001 (02.10.2001) US

(71) Applicant *(for all designated States except US)*: **NETWORKS ASSOCIATES TECHNOLOGY, INC.** [US/US]; 3965 Freedom Circle, Santa Clara, CA 95054 (US).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: **SINGLETON, Richard, B.** [GB/GB]; 4 Coral Close, Eaton Bray, Bedfordshire LU6 2AS (GB).

(74) Agents: **MALLIE, Michael, J.** et al.; Blakely, Sokoloff, Taylor & Zafman LLP, 12400 Wilshire Boulevard, 7th floor, Los Angeles, CA 90025 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

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

WO 03/029940 A2

(54) Title: MASTER SECURITY POLICY SERVER

(57) Abstract: A master policy server manages security polices for client computers through a network of local policy servers. Each local policy server is responsible for the security policies on a group of clients and maintains a data store containing the security policies and security information pertaining to the clients. Periodically, the master policy server and the local policy server synchronize, at which time the master policy server replicates updated policies to the local policy servers and the policy servers upload client security statistics to the master policy server for consolidation into a global status.

ForcePoint LLC
EX-1009
Page314

MASTER SECURITY POLICY SERVER

FIELD OF THE INVENTION

This invention relates generally to computer security, and more particularly to managing security policies through a centralized server.

COPYRIGHT NOTICE/PERMISSION

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever. The following notice applies to the software and data as described below and in the drawings hereto: Copyright © 2001, Networks Associates Technology, Inc., All Rights Reserved.

BACKGROUND OF THE INVENTION

Organizations often manage their computer security policies from a central location, typically employing a single computer server to manage the security polices on networked user (client) computers. The clients poll the server several times a day to check for, and optionally download, updated security policies and to upload their status to the server. Assuming a client and the server exchange a large amount of data several times a day, the data traffic between the server and even a small number clients can cause significant degradation for overall network communications.

SUMMARY OF THE INVENTION

A master policy server manages security polices for client computers through a network of local policy servers. Each local policy server is responsible for the security policies on a group of clients and maintains a data store containing the security policies and security information pertaining to the clients. Periodically, the master policy server and the local policy server synchronize, at which time the master policy server replicates

updated policies to the local policy servers and the local policy servers upload client security statistics to the master policy server for consolidation into a global status. A local policy server may also request an updated security policy outside of the synchronization timeframe. Similarly, the master policy server may also request the client statistics from a local policy server outside of the synchronization timeframe.

Because the local policy servers consolidate the statistics from the clients prior to uploading it to the master policy server, the amount of data flowing through the network to the master policy server is greatly reduced. Similarly, because the master policy server replicates the security policies to a few local policy servers instead of to each client, the amount of data flowing through the network from the master policy server is also reduced.

The present invention describes systems, clients, servers, methods, and computer-readable media of varying scope. In addition to the aspects and advantages of the present invention described in this summary, further aspects and advantages of the invention will become apparent by reference to the drawings and by reading the detailed description that follows.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a diagram illustrating a system-level overview of an embodiment of the invention;

Figure 2A is a flowchart of a method to be performed by a master server according to an embodiment of the invention;

Figure 2B is a flowchart of a method to be performed by a local server operating in conjunction with the master server of Figure 2A;

Figure 3A is a diagram of one embodiment of an operating environment suitable for practicing the present invention; and

Figure 3B is a diagram of one embodiment of a computer system suitable for use in the operating environment of Figure 3A.

ForcePoint LLC
EX-1009
Page316

DETAILED DESCRIPTION OF THE INVENTION

In the following detailed description of embodiments of the invention, reference is made to the accompanying drawings in which like references indicate similar elements, and in which is shown by way of illustration specific embodiments in which the invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention, and it is to be understood that other embodiments may be utilized and that logical, mechanical, electrical, functional, and other changes may be made without departing from the scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims.

A system level overview of the operation of an embodiment of the invention is described by reference to Figure 1, which illustrates a security policy distribution system 100. The system 100 utilizes a master policy server 101 to manage security policies on client (user) computers through a network 129 of local policy severs A 103, B 105 and C 107. For example, local policy server A103 manages client A-1 115 through client A-N 117, while local policy server B 105 manages client B-1 119 through B-N 121. Although the clients are represented as individual systems in Figure 1, it will be appreciated that they may be grouped together by hardware and software platform type, domain name, site location, or physical or logical region.

Each local policy server has a local data store 109, 111, 113 that contains the security policies and security information collected from the client computers it manages. Each type of hardware and software platform acting as a client computer may be associated with a exemplary security policy or may share exemplary security policies with other platforms. The security policy may contain configuration parameters for anti-virus programs, firewalls, and other security software that protect a client computer from compromise by a third-party.

Communication between the local policy servers 103, 105, 107 and the master policy server 101 through network 129 is intermittent. Each local policy server 103, 105, 107 is responsible for periodically querying the master policy server 101 to determine if the security policies applicable to its clients have changed. The local policy servers also periodically, or upon request, send client security statistics derived from the security

ForcePoint LLC
EX-1009
Page317

information on local data stores 109, 111, 113 to the master policy server 101, which acts as a consolidation point for status information regarding the overall security of the system 100. The statistics from the local policy servers are stored in a global data store 127. When a global status for the system 100 is requested, the master policy server 101 derives the status from the statistics in the global data store 127 and, optionally, from additional statistics obtained from the local policy servers. More detailed status information for particular clients or groups of clients is obtained from the appropriate local policy server.

In one embodiment, the master policy server 101 and the local policy servers 103, 105, 107 synchronize security policies and statistics at times when less data traffic is generally experienced on the network 129. When the local policy servers are physically located in different time zones, the synchronization may occur at several points during a twenty-four hour period. In an alternate embodiment, the local policy servers can schedule checks for updated policies in addition to the synchronization process. Furthermore, it will be appreciated that the synchronization at a local policy server may happen more than once a day. The network 129 connecting the master policy server and the local policy servers is secured using any of several well-known secure transmission protocols when the security policies are being uploaded to the master policy server 101 or replicated to the local policy servers 103, 105, 107. Otherwise, no particular network transmission protocols are required in the system 100.

When the system 100 is installed, the system administrator may create the initial security policies at one of the local policy servers 103, 105, 107 for transfer to the master policy server 101 and subsequent replication to the other local policy servers, or directly at the master policy server 101. Similarly, updates to the security policies may be performed at a local policy server or at the master policy server. In one embodiment, the master policy server 101 maintains global level security policy configurations and the local policy servers 103, 105, 107 derive their local level configuration and set-up policies for their clients from the global level configurations.

The number of local policy servers is dependent upon the number of clients at each site and the physical locations of the sites. Because the master policy server 101 only sends and receives data from the local policy servers 103, 105, 107 instead of each

ForcePoint LLC
EX-1009
Page318

of the clients, a single master policy server and common TCP/IP wide-area networks are generally able to handle the amount of data being transferred in the system 100. Alternate embodiments in which additional levels of servers are incorporated between the local policy servers 103,105, 107 and the master policy server 101 are also contemplated and are considered within the scope of the invention.

The operations of an embodiment of a security policy distribution system 100 have been described in terms of a single master policy server and three local policy servers as illustrated in Figure 1, but the invention is not so limited. Next, the particular methods of the invention that perform the operations for the system 100 are described in terms of computer software with reference to a series of flowcharts. The methods to be performed by a computer constitute computer programs made up of computer-executable instructions illustrated as blocks (acts). Describing the methods by reference to a flowchart enables one skilled in the art to develop such programs including such instructions to carry out the methods on suitably configured computers (the processing unit of the computer executing the instructions from computer-readable media). The computer-executable instructions may be written in a computer programming language or may be embodied in firmware logic. If written in a programming language conforming to a recognized standard, such instructions can be executed on a variety of hardware platforms and for interface to a variety of operating systems. In addition, the present invention is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the invention as described herein. Furthermore, it is common in the art to speak of software, in one form or another (e.g., program, procedure, process, application, module, logic...), as taking an action or causing a result. Such expressions are merely a shorthand way of saying that execution of the software by a computer causes the processor of the computer to perform an action or a produce a result.

Referring first to Figure 2A, the acts to be performed by a computer executing a master server method 200 to perform the operations described for the master policy server 101 in Figure 1 is shown. The master server method 200 is invoked by one or more of a series of predetermined events. If a new policy has been created, either at the master policy server 101, or at one of the local policy servers, 103, 105, 107, (block 201),

the master server method 200 obtains and stores the security policy at block 203. If the master server method 200 receives a request for a new policy from a local policy server (block 205), the master server method 200 replicates the policy to the requestor at block 207. It will be appreciated that the master policy server will replicate those policies which are requested by the local policy server, i.e., those policies particular to the client platforms which the local policy server is managing. If the master server method 200 receives a request for system status (block 209), the master server method 200 determines if the request is for historical or current status (block 213). If the report is for current status, the master server method 200 obtains the current statistics from the local servers at block 215. The appropriate status is returned to the requester at the block 217. Otherwise, the event that invoked the master server method 200 is a scheduled synchronization event and the master server method 200 synchronizes security policies and statistics with the appropriate local policy servers at block 211.

A local server method 230 is illustrated in Figure 2B that performs the operations previously described for the local policy servers 103, 105, 107 in Figure 1. As with the master server method 200, the local server method 230 is invoked by one or more of a predetermined sequence of events. If a new policy has been configured on the local policy server (block 231), the local server method 230 sends the new policy to the master policy server at block 233 for replication to the other local policy servers. If the event is a scheduled check for the availability of new policies (block 235), the local server method 230 requests appropriate new policies from the master policy server at block 237 and receive and apply any new policies at block 239. If the local server method 230 receives a request for current status from the master server method 200 (block 241), it send its current statistics to the master policy server at block 243. Otherwise, the event is a scheduled synchronization event and the local server method 230 synchronizes with the master policy server at block 245, sending statistics from the local data store to the master policy server and receiving any updates to the security policies.

The methods performed by a master policy server and local policy server have been shown by reference to flowcharts in Figures 2A and 2B, respectively, including all the acts from 201 until 217 and from 231 until 245. It will be appreciated that more or fewer processes may be incorporated into the methods illustrated in Figures 2A-B

ForcePoint LLC
EX-1009
Page320

without departing from the scope of the invention, and that no particular order is implied by the arrangement of blocks shown and described herein.

The following description of Figures 3A-B is intended to provide an overview of computer hardware and other operating components suitable for performing the methods of the invention described above, but is not intended to limit the applicable environments. One of skill in the art will immediately appreciate that the invention can be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The invention can also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network.

Figure 3A shows several computer systems that are coupled together through a network 3, such as the Internet. The term "Internet" as used herein refers to a network of networks which uses certain protocols, such as the TCP/IP protocol, and possibly other protocols such as the hypertext transfer protocol (HTTP) for hypertext markup language (HTML) documents that make up the World Wide Web (web). The physical connections of the Internet and the protocols and communication procedures of the Internet are well known to those of skill in the art. Access to the Internet 3 is typically provided by Internet service providers (ISP), such as the ISPs 5 and 7. Users on client systems, such as client computer systems 21, 25, 35, and 37 obtain access to the Internet through the Internet service providers, such as ISPs 5 and 7. Access to the Internet allows users of the client computer systems to exchange information, receive and send e-mails, and view documents, such as documents which have been prepared in the HTML format. These documents are often provided by web servers, such as web server 9 which is considered to be "on" the Internet. Often these web servers are provided by the ISPs, such as ISP 5, although a computer system can be set up and connected to the Internet without that system being also an ISP as is well known in the art.

The web server 9 is typically at least one computer system which operates as a server computer system and is configured to operate with the protocols of the World Wide Web and is coupled to the Internet. Optionally, the web server 9 can be part of an ISP which provides access to the Internet for client systems. The web server 9 is shown

ForcePoint LLC
EX-1009
Page321

coupled to the server computer system 11 which itself is coupled to web content 10, which can be considered a form of a media database. It will be appreciated that while two computer systems 9 and 11 are shown in Figure 3A, the web server system 9 and the server computer system 11 can be one computer system having different software components providing the web server functionality and the server functionality provided by the server computer system 11 which will be described further below.

Client computer systems 21, 25, 35, and 37 can each, with the appropriate web browsing software, view HTML pages provided by the web server 9. The ISP 5 provides Internet connectivity to the client computer system 21 through the modem interface 23 which can be considered part of the client computer system 21. The client computer system can be a personal computer system, a network computer, a Web TV system, or other such computer system. Similarly, the ISP 7 provides Internet connectivity for client systems 25, 35, and 37, although as shown in Figure 3A, the connections are not the same for these three computer systems. Client computer system 25 is coupled through a modem interface 27 while client computer systems 35 and 37 are part of a LAN. While Figure 3A shows the interfaces 23 and 27 as generically as a "modem," it will be appreciated that each of these interfaces can be an analog modem, ISDN modem, cable modem, satellite transmission interface (e.g. "Direct PC"), or other interfaces for coupling a computer system to other computer systems. Client computer systems 35 and 37 are coupled to a LAN 33 through network interfaces 39 and 41, which can be Ethernet network or other network interfaces. The LAN 33 is also coupled to a gateway computer system 31 which can provide firewall and other Internet related services for the local area network. This gateway computer system 31 is coupled to the ISP 7 to provide Internet connectivity to the client computer systems 35 and 37. The gateway computer system 31 can be a conventional server computer system. Also, the web server system 9 can be a conventional server computer system.

Alternatively, as well-known, a server computer system 43 can be directly coupled to the LAN 33 through a network interface 45 to provide files 47 and other services to the clients 35, 37, without the need to connect to the Internet through the gateway system 31.

Figure 3B shows one example of a conventional computer system that can be used as a client computer system or a server computer system or as a web server system. It will also be appreciated that such a computer system can be used to perform many of the functions of an Internet service provider, such as ISP 5. The computer system 51 interfaces to external systems through the modem or network interface 53. It will be appreciated that the modem or network interface 53 can be considered to be part of the computer system 51. This interface 53 can be an analog modem, ISDN modem, cable modem, token ring interface, satellite transmission interface (e.g. "Direct PC"), or other interfaces for coupling a computer system to other computer systems. The computer system 51 includes a processing unit 55, which can be a conventional microprocessor such as an Intel Pentium microprocessor or Motorola Power PC microprocessor. Memory 59 is coupled to the processor 55 by a bus 57. Memory 59 can be dynamic random access memory (DRAM) and can also include static RAM (SRAM). The bus 57 couples the processor 55 to the memory 59 and also to non-volatile storage 65 and to display controller 61 and to the input/output (I/O) controller 67. The display controller 61 controls in the conventional manner a display on a display device 63 which can be a cathode ray tube (CRT) or liquid crystal display. The input/output devices 69 can include a keyboard, disk drives, printers, a scanner, and other input and output devices, including a mouse or other pointing device. The display controller 61 and the I/O controller 67 can be implemented with conventional well known technology. A digital image input device 71 can be a digital camera which is coupled to the I/O controller 67 in order to allow images from the digital camera to be input into the computer system 51. The non-volatile storage 65 is often a magnetic hard disk, an optical disk, or another form of storage for large amounts of data. Some of this data is often written, by a direct memory access process, into memory 59 during execution of software in the computer system 51. One of skill in the art will immediately recognize that the term "computer-readable medium" includes any type of storage device that is accessible by the processor 55 and also encompasses a carrier wave that encodes a data signal.

It will be appreciated that the computer system 51 is one example of many possible computer systems which have different architectures. For example, personal computers based on an Intel microprocessor often have multiple buses, one of which can

be an input/output (I/O) bus for the peripherals and one that directly connects the processor 55 and the memory 59 (often referred to as a memory bus). The buses are connected together through bridge components that perform any necessary translation due to differing bus protocols.

Network computers are another type of computer system that can be used with the present invention. Network computers do not usually include a hard disk or other mass storage, and the executable programs are loaded from a network connection into the memory 59 for execution by the processor 55. A Web TV system, which is known in the art, is also considered to be a computer system according to the present invention, but it may lack some of the features shown in Figure 3B, such as certain input or output devices. A typical computer system will usually include at least a processor, memory, and a bus coupling the memory to the processor.

It will also be appreciated that the computer system 51 is controlled by operating system software which includes a file management system, such as a disk operating system, which is part of the operating system software. One example of an operating system software with its associated file management system software is the family of operating systems known as Windows® from Microsoft Corporation of Redmond, Washington, and their associated file management systems. The file management system is typically stored in the non-volatile storage 65 and causes the processor 55 to execute the various acts required by the operating system to input and output data and to store data in memory, including storing files on the non-volatile storage 65.

A security policy distribution system that is managed by a master security policy server has been described. Although specific embodiments have been illustrated and described herein, it will be appreciated by those of ordinary skill in the art that any arrangement which is calculated to achieve the same purpose may be substituted for the specific embodiments shown. This application is intended to cover any adaptations or variations of the present invention.

The terminology used in this application with respect to network communications is meant to include all communication media and environments, including local and wide area networks, public and private communications environments, and wired and wireless

communications media. Therefore, it is manifestly intended that this invention be limited only by the following claims and equivalents thereof.

ForcePoint LLC
EX-1009
Page325

CLAIMS

What is claimed is:

1. A computerized method of distributing security policies comprising:
maintaining a security policy at a master policy server; and
periodically synchronizing the master policy server and a local policy server to replicate the security policy at the local policy server.

2. The computerized method of claim 1, wherein the synchronizing further comprises obtaining security statistics from the local policy server by the master policy server.

3. The computerized method of claim 2 further comprising:
deriving a global status from the statistics.

4. The computerized method of claim 1, further comprising:
obtaining security statistics by the master policy server upon request to the local policy server.

5. The computerized method of claim 1 further comprising:
replicating the security policy to the local policy server upon request to the master policy server.

6. The computerized method of claim 1 further comprising:
creating the security policy at the local policy server; and
transferring the security policy to the master policy server.

7. The computerized method of claim 1 further comprising:
creating the security policy at the master policy server.

ForcePoint LLC
EX-1009
Page326

8. The computerized method of claim 1, wherein the synchronizing is performed securely across a communications medium coupling the master policy server and the local policy server.

9. The computerized method of claim 1, further comprising:

managing security for a plurality of client platforms by the local policy server, the security policy comprising security parameters particular to each client platform.

10. The computerized method of claim 9, further comprising:

deriving the security policy parameters particular to each client platform from global security parameters, the security policy at the master policy comprising the global security parameters.

11. A computer-readable medium having executable instructions to cause a computer to perform a method comprising:

maintaining a security policy at a master policy server; and

periodically synchronizing the master policy server and a local policy server to replicate the security policy at the local policy server.

12. The computer-readable medium of claim 11, wherein the synchronizing further comprises obtaining security statistics from the local policy server by the master policy server.

13. The computer-readable medium of claim 12, wherein the method further comprises:

deriving a global status from the statistics.

14. The computer-readable medium of claim 11, wherein the method further comprises:

obtaining security statistics by the master policy server upon request to the local policy server.

15. The computer-readable medium of claim 11, wherein the method further comprises:

replicating the security policy to the local policy server upon request to the master policy server.

16. The computer-readable medium of claim 11, wherein the method further comprises:

creating the security policy at the local policy server; and

transferring the security policy to the master policy server.

17. The computer-readable medium of claim 11, wherein the method further comprises:

creating the security policy at the master policy server.

18. The computer-readable medium of claim 11, wherein the synchronizing is performed securely across a communications medium coupling the master policy server and the local policy server.

19. The computer-readable medium of claim 11, wherein the method further comprises:

managing security for a plurality of client platforms by the local policy server, the security policy comprising security parameters particular to each client platform.

20. The computer-readable medium of claim 19, wherein the method further comprises:

deriving the security policy parameters particular to each client platform from global security parameters, the security policy at the master policy comprising the global security parameters.

21. A computer system comprising:

a processor coupled to a memory through a bus;

a network interface coupled to the processor through the bus; and

ForcePoint LLC
EX-1009
Page328

a master server process executed from the memory by the processor to cause the processor to maintain a security policy and to periodically synchronize with a local policy server through the network interface to replicate the security policy at the local policy server.

22. The computer system of claim 21, wherein the master server process further causes the processor to obtain security statistics from the local policy server through the network interface during synchronization.

23. The computer system of claim 22, wherein the master server process further causes the processor to derive a global status from the statistics.

24. The computer system of claim 21, wherein the master server process further causes the processor to request security statistics from the local policy server through the network interface.

25. The computer system of claim 21, wherein the master server process further causes the processor to receive a request from the local policy server and to replicate the security policy to the local policy server in response.

26. The computer system of claim 21, wherein the master server process further causes the processor to create the security policy.

27. The computer system of claim 21, wherein the master server process further causes the processor to couple the network interface to a secure communications medium for synchronization.

28. A computer system comprising:

a processor coupled to a memory through a bus;

a network interface coupled to the processor through the bus; and

a local server process executed from the memory by the processor to cause the processor to periodically synchronize with a master policy server through the network interface to receive a security policy from a master policy server.

29. The computer system of claim 28, wherein the local server process further causes the processor to transfer security statistics to the master policy server through the network interface during synchronization.

30. The computer system of claim 28, wherein the local server process further causes the processor to receive a request for security statistics from the master policy server and to transfer the security statistics to the master policy server through the network interface in response.

31. The computer system of claim 28, wherein the local server process further causes the processor to request a security policy from the master policy server through the network interface and to receive the security policy in response.

32. The computer system of claim 28, wherein the local server process further causes the processor to create the security policy and to transfer the security policy to the master policy server through the network interface.

33. The computer system of claim 28, wherein the local server process further causes the processor to manage security for a plurality of client platforms, the security policy comprising security parameters particular to each client platform.

34. The computer system of claim 33, wherein the local server process further causes the processor to derive the security policy parameters particular to each client platform from global security parameters, the security policy at the master policy server comprising the global security parameters.

35. An apparatus comprising:

network means for interfacing to a network; and

ForcePoint LLC
EX-1009
Page330

master policy means for maintaining a security policy and for periodically synchronizing with a local policy means through the network interface to replicate the security policy at the local policy means.

36. The apparatus of claim 35, wherein the master policy means is further operable for obtaining security statistics from the local policy means through the network means during synchronization.

37. The apparatus of claim 36, wherein the master policy means is further operable for deriving a global status from the statistics.

38. The apparatus of claim 35, wherein the master policy means is further operable for requesting security statistics from the local policy means through the network means.

39. The apparatus of claim 35, wherein the master policy means is further operable for receiving a request from the local policy means through the network means and for replicating the security policy to the local policy means through the network means in response.

40. The apparatus of claim 35, wherein the master policy means is further operable for creating the security policy.

41. The apparatus of claim 35, wherein the network means is further operable for coupling to a secure communications medium for synchronization between the master policy means and the local policy means.

42. An apparatus comprising:

network means for interfacing to a network; and

local policy means for periodically synchronizing to a master policy means through the network means to receive a security policy form the master policy means.

43. The apparatus of claim 42, wherein the local policy means is further operable for transferring security statistics to the master policy means through the network means during synchronization.

44. The apparatus of claim 42, wherein the local policy means is further operable for receiving a request for security statistics from the master policy means through the network means and for transferring the security statistics to the master policy means through the network means in response.

45. The apparatus of claim 42, wherein the local policy means is further operable for requesting a security policy from the master policy means through the network means and for receiving the security policy from the master policy means through the network means in response.

46. The apparatus of claim 42, wherein the local policy means is further operable for creating the security policy and for transferring the security policy to the master policy means through the network means.

47. The computer system of claim 42, wherein the local security means is further operable for managing security for a plurality of client platforms, the security policy comprising security parameters particular to each client platform.

48. The computer system of claim 47, wherein the local security means is further operable for deriving the security policy parameters particular to each client platform from global security parameters, the security policy at the master policy means comprising the global security parameters.

ForcePoint LLC
EX-1009
Page332

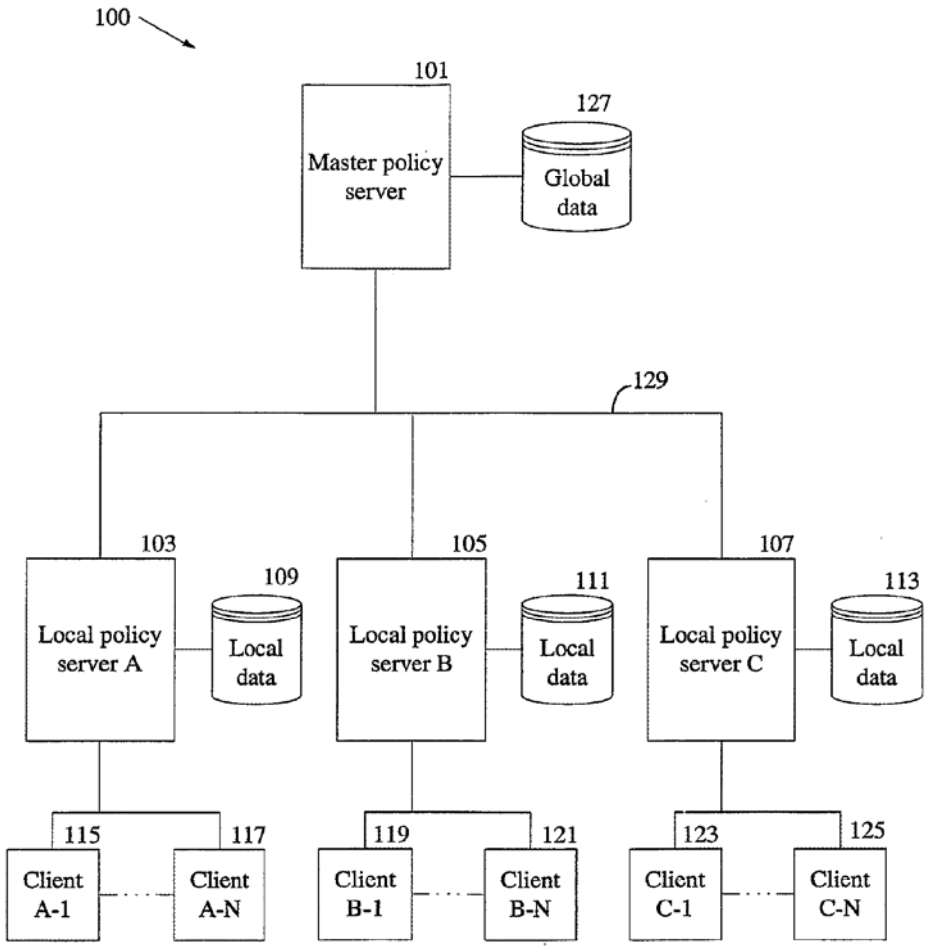

100
101
Master policy server
127
Global data
129
103
Local policy server A
109
Local data
105
Local policy server B
111
Local data
107
Local policy server C
113
Local data
115
Client A-1
117
Client A-N
119
Client B-1
121
Client B-N
123
Client C-1
125
Client C-N

Figure 1

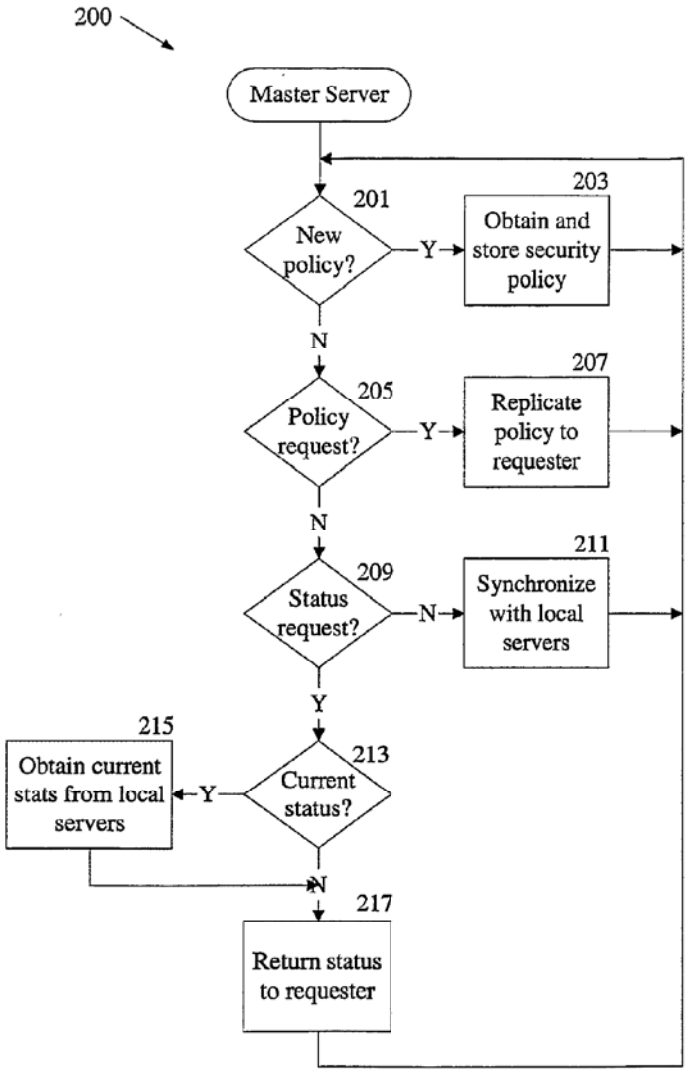


Figure 2A

ForcePoint LLC
EX-1009
Page334

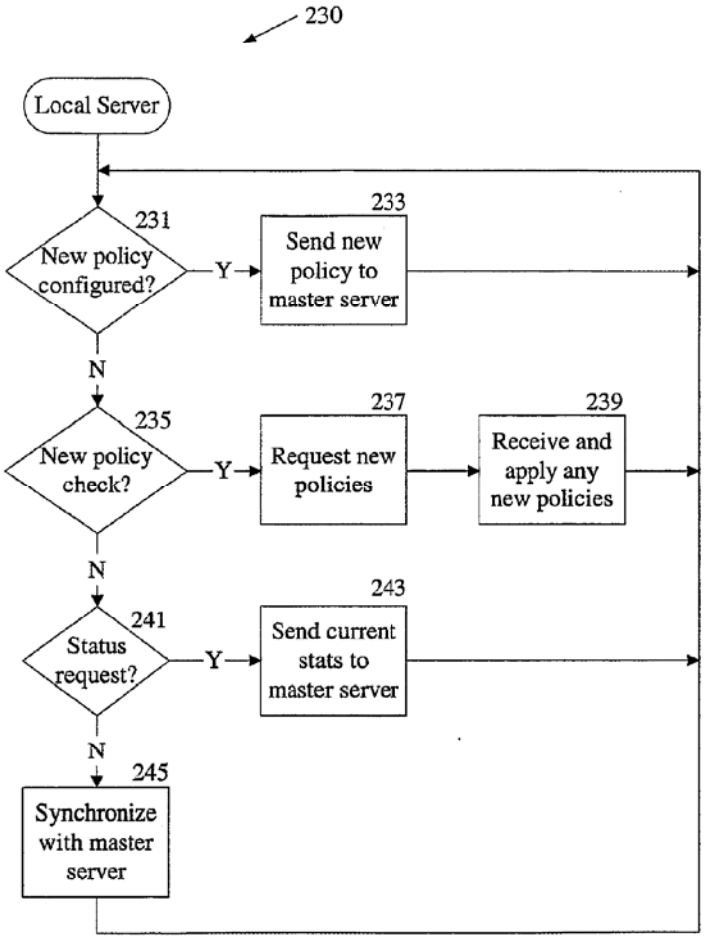


Figure 2B

ForcePoint LLC
EX-1009
Page335

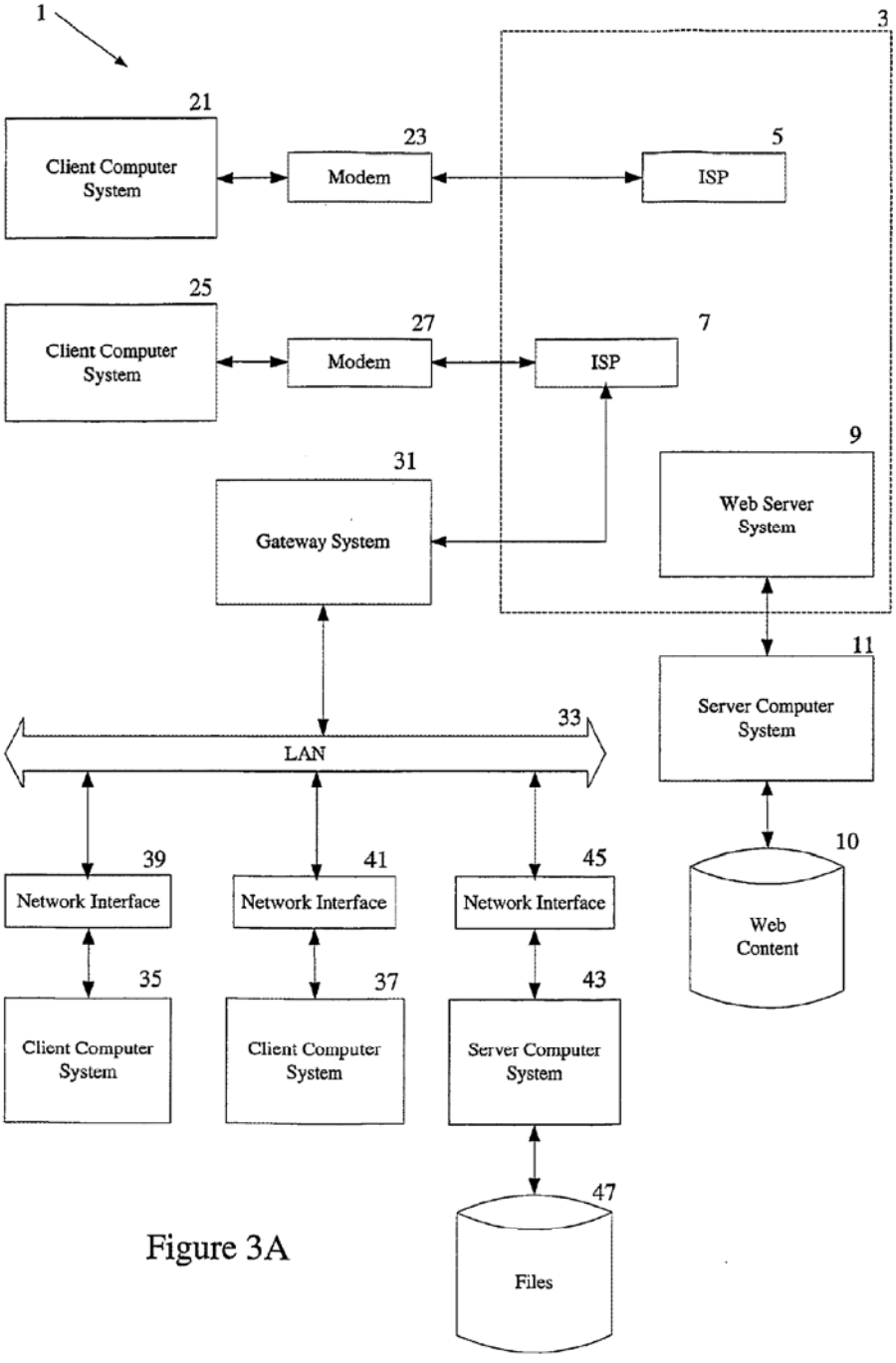

1
3
21
Client Computer System
23
Modem
5
ISP
25
Client Computer System
27
Modem
7
ISP
9
Web Server System
31
Gateway System
11
Server Computer System
33
LAN
10
Web Content
39
Network Interface
41
Network Interface
45
Network Interface
35
Client Computer System
37
Client Computer System
43
Server Computer System
47
Files

Figure 3A

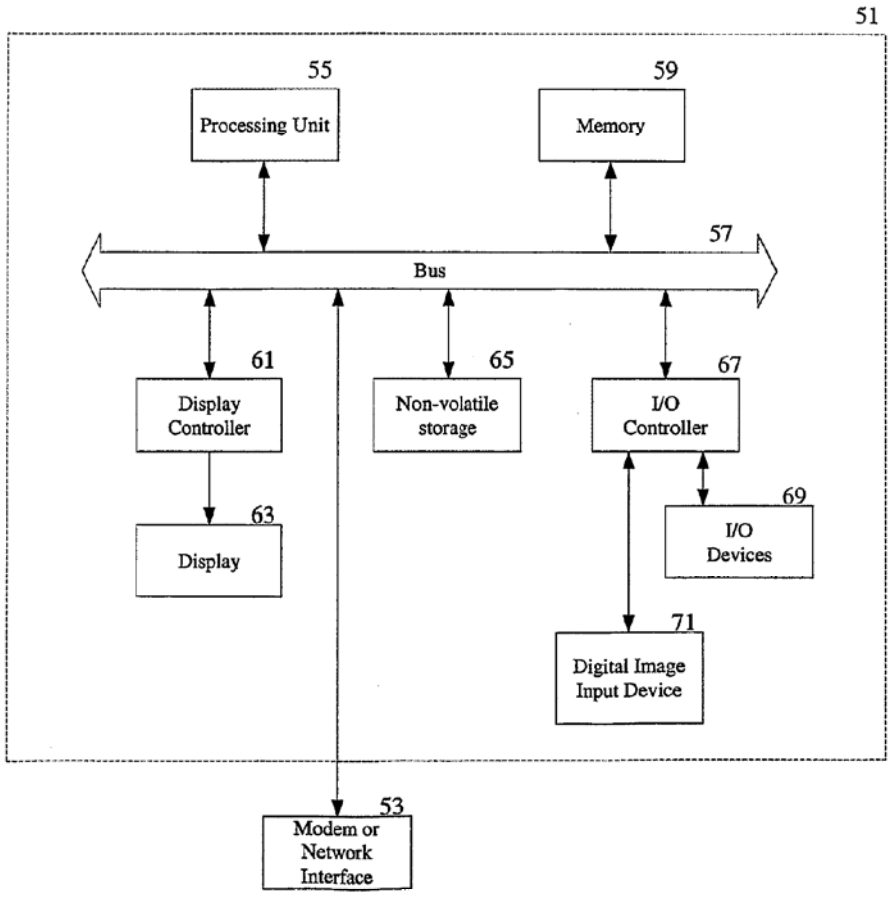

51
55
Processing Unit
59
Memory
57
Bus
61
Display Controller
65
Non-volatile storage
67
I/O Controller
63
Display
69
I/O Devices
71
Digital Image Input Device
53
Modem or Network Interface

Figure 3B

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 April 2003 (10.04.2003)

PCT

(10) International Publication Number
WO 03/029941 A2

(51) International Patent Classification[7]: G06F 1/00

(21) International Application Number: PCT/US02/28513

(22) International Filing Date:
6 September 2002 (06.09.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/969,686 2 October 2001 (02.10.2001) US

(71) Applicant: **NETWORKS ASSOCIATES TECHNOLOGY, INC.** [US/US]; 3965 Freedom Circle, Santa Clara, CA 95054 (US).

(72) Inventor: **SINGLETON, Richard, B.**; 4 Coral Close, Eaton Bray, Bedfordshire LU6 2AS (GB).

(74) Agents: **MALLIE, Michael, J.** et al.; Blakely, Sokoloff, Taylor & Zafman LLP, 7th floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

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

WO 03/029941 A2

(54) Title: MASTER SECURITY POLICY SERVER

(57) Abstract: A master policy server manages security polices for client computers through a network of local policy servers. Each local policy server is responsible for the security policies on a group of clients and maintains a data store containing the security policies and security information pertaining to the clients. Periodically, the master policy server and the local policy server synchronize, at which time the master policy server replicates updated policies to the local policy servers and the local policy servers upload client security statistics to the master policy server for consolidation into a global status.

ForcePoint LLC
EX-1009
Page338

MASTER SECURITY POLICY SERVER

FIELD OF THE INVENTION

This invention relates generally to computer security, and more particularly to managing security policies through a centralized server.

COPYRIGHT NOTICE/PERMISSION

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever. The following notice applies to the software and data as described below and in the drawings hereto: Copyright © 2001, Networks Associates Technology, Inc., All Rights Reserved.

BACKGROUND OF THE INVENTION

Organizations often manage their computer security policies from a central location, typically employing a single computer server to manage the security polices on networked user (client) computers. The clients poll the server several times a day to check for, and optionally download, updated security policies and to upload their status to the server. Assuming a client and the server exchange a large amount of data several times a day, the data traffic between the server and even a small number clients can cause significant degradation for overall network communications.

SUMMARY OF THE INVENTION

A master policy server manages security polices for client computers through a network of local policy servers. Each local policy server is responsible for the security policies on a group of clients and maintains a data store containing the security policies and security information pertaining to the clients. Periodically, the master policy server and the local policy server synchronize, at which time the master policy server replicates

updated policies to the local policy servers and the local policy servers upload client security statistics to the master policy server for consolidation into a global status. A local policy server may also request an updated security policy outside of the synchronization timeframe. Similarly, the master policy server may also request the client statistics from a local policy server outside of the synchronization timeframe.

Because the local policy servers consolidate the statistics from the clients prior to uploading it to the master policy server, the amount of data flowing through the network to the master policy server is greatly reduced. Similarly, because the master policy server replicates the security policies to a few local policy servers instead of to each client, the amount of data flowing through the network from the master policy server is also reduced.

The present invention describes systems, clients, servers, methods, and computer-readable media of varying scope. In addition to the aspects and advantages of the present invention described in this summary, further aspects and advantages of the invention will become apparent by reference to the drawings and by reading the detailed description that follows.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a diagram illustrating a system-level overview of an embodiment of the invention;

Figure 2A is a flowchart of a method to be performed by a master server according to an embodiment of the invention;

Figure 2B is a flowchart of a method to be performed by a local server operating in conjunction with the master server of Figure 2A;

Figure 3A is a diagram of one embodiment of an operating environment suitable for practicing the present invention; and

Figure 3B is a diagram of one embodiment of a computer system suitable for use in the operating environment of Figure 3A.

ForcePoint LLC
EX-1009
Page340

DETAILED DESCRIPTION OF THE INVENTION

In the following detailed description of embodiments of the invention, reference is made to the accompanying drawings in which like references indicate similar elements, and in which is shown by way of illustration specific embodiments in which the invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention, and it is to be understood that other embodiments may be utilized and that logical, mechanical, electrical, functional, and other changes may be made without departing from the scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims.

A system level overview of the operation of an embodiment of the invention is described by reference to Figure 1, which illustrates a security policy distribution system 100. The system 100 utilizes a master policy server 101 to manage security policies on client (user) computers through a network 129 of local policy severs A 103, B 105 and C 107. For example, local policy server A103 manages client A-1 115 through client A-N 117, while local policy server B 105 manages client B-1 119 through B-N 121. Although the clients are represented as individual systems in Figure 1, it will be appreciated that they may be grouped together by hardware and software platform type, domain name, site location, or physical or logical region.

Each local policy server has a local data store 109, 111, 113 that contains the security policies and security information collected from the client computers it manages. Each type of hardware and software platform acting as a client computer may be associated with a exemplary security policy or may share exemplary security policies with other platforms. The security policy may contain configuration parameters for anti-virus programs, firewalls, and other security software that protect a client computer from compromise by a third-party.

Communication between the local policy servers 103, 105, 107 and the master policy server 101 through network 129 is intermittent. Each local policy server 103, 105, 107 is responsible for periodically querying the master policy server 101 to determine if the security policies applicable to its clients have changed. The local policy servers also periodically, or upon request, send client security statistics derived from the security

ForcePoint LLC
EX-1009
Page341

information on local data stores 109, 111, 113 to the master policy server 101, which acts as a consolidation point for status information regarding the overall security of the system 100. The statistics from the local policy servers are stored in a global data store 127. When a global status for the system 100 is requested, the master policy server 101 derives the status from the statistics in the global data store 127 and, optionally, from additional statistics obtained from the local policy servers. More detailed status information for particular clients or groups of clients is obtained from the appropriate local policy server.

In one embodiment, the master policy server 101 and the local policy servers 103, 105, 107 synchronize security policies and statistics at times when less data traffic is generally experienced on the network 129. When the local policy servers are physically located in different time zones, the synchronization may occur at several points during a twenty-four hour period. In an alternate embodiment, the local policy servers can schedule checks for updated policies in addition to the synchronization process. Furthermore, it will be appreciated that the synchronization at a local policy server may happen more than once a day. The network 129 connecting the master policy server and the local policy servers is secured using any of several well-known secure transmission protocols when the security policies are being uploaded to the master policy server 101 or replicated to the local policy servers 103, 105, 107. Otherwise, no particular network transmission protocols are required in the system 100.

When the system 100 is installed, the system administrator may create the initial security policies at one of the local policy servers 103, 105, 107 for transfer to the master policy server 101 and subsequent replication to the other local policy servers, or directly at the master policy server 101. Similarly, updates to the security policies may be performed at a local policy server or at the master policy server. In one embodiment, the master policy server 101 maintains global level security policy configurations and the local policy servers 103, 105, 107 derive their local level configuration and set-up policies for their clients from the global level configurations.

The number of local policy servers is dependent upon the number of clients at each site and the physical locations of the sites. Because the master policy server 101 only sends and receives data from the local policy servers 103, 105, 107 instead of each

ForcePoint LLC
EX-1009
Page342

of the clients, a single master policy server and common TCP/IP wide-area networks are generally able to handle the amount of data being transferred in the system 100. Alternate embodiments in which additional levels of servers are incorporated between the local policy servers 103,105, 107 and the master policy server 101 are also contemplated and are considered within the scope of the invention.

The operations of an embodiment of a security policy distribution system 100 have been described in terms of a single master policy server and three local policy servers as illustrated in Figure 1, but the invention is not so limited. Next, the particular methods of the invention that perform the operations for the system 100 are described in terms of computer software with reference to a series of flowcharts. The methods to be performed by a computer constitute computer programs made up of computer-executable instructions illustrated as blocks (acts). Describing the methods by reference to a flowchart enables one skilled in the art to develop such programs including such instructions to carry out the methods on suitably configured computers (the processing unit of the computer executing the instructions from computer-readable media). The computer-executable instructions may be written in a computer programming language or may be embodied in firmware logic. If written in a programming language conforming to a recognized standard, such instructions can be executed on a variety of hardware platforms and for interface to a variety of operating systems. In addition, the present invention is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the invention as described herein. Furthermore, it is common in the art to speak of software, in one form or another (e.g., program, procedure, process, application, module, logic...), as taking an action or causing a result. Such expressions are merely a shorthand way of saying that execution of the software by a computer causes the processor of the computer to perform an action or a produce a result.

Referring first to Figure 2A, the acts to be performed by a computer executing a master server method 200 to perform the operations described for the master policy server 101 in Figure 1 is shown. The master server method 200 is invoked by one or more of a series of predetermined events. If a new policy has been created, either at the master policy server 101, or at one of the local policy servers, 103, 105, 107, (block 201),

ForcePoint LLC
EX-1009
Page343

the master server method 200 obtains and stores the security policy at block 203. If the master server method 200 receives a request for a new policy from a local policy server (block 205), the master server method 200 replicates the policy to the requestor at block 207. It will be appreciated that the master policy server will replicate those policies which are requested by the local policy server, i.e., those policies particular to the client platforms which the local policy server is managing. If the master server method 200 receives a request for system status (block 209), the master server method 200 determines if the request is for historical or current status (block 213). If the report is for current status, the master server method 200 obtains the current statistics from the local servers at block 215. The appropriate status is returned to the requester at the block 217. Otherwise, the event that invoked the master server method 200 is a scheduled synchronization event and the master server method 200 synchronizes security policies and statistics with the appropriate local policy servers at block 211.

A local server method 230 is illustrated in Figure 2B that performs the operations previously described for the local policy servers 103, 105, 107 in Figure 1. As with the master server method 200, the local server method 230 is invoked by one or more of a predetermined sequence of events. If a new policy has been configured on the local policy server (block 231), the local server method 230 sends the new policy to the master policy server at block 233 for replication to the other local policy servers. If the event is a scheduled check for the availability of new policies (block 235), the local server method 230 requests appropriate new policies from the master policy server at block 237 and receive and apply any new policies at block 239. If the local server method 230 receives a request for current status from the master server method 200 (block 241), it send its current statistics to the master policy server at block 243. Otherwise, the event is a scheduled synchronization event and the local server method 230 synchronizes with the master policy server at block 245, sending statistics from the local data store to the master policy server and receiving any updates to the security policies.

The methods performed by a master policy server and local policy server have been shown by reference to flowcharts in Figures 2A and 2B, respectively, including all the acts from 201 until 217 and from 231 until 245. It will be appreciated that more or fewer processes may be incorporated into the methods illustrated in Figures 2A-B

ForcePoint LLC
EX-1009
Page344