(19) Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 603 11 666 T2** 2007.11.22

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 1 543 396 B1**
(21) Deutsches Aktenzeichen: **603 11 666.3**
(86) PCT-Aktenzeichen: **PCT/GB03/03112**
(96) Europäisches Aktenzeichen: **03 738 350.2**
(87) PCT-Veröffentlichungs-Nr.: **WO 2004/010269**
(86) PCT-Anmeldetag: **17.07.2003**
(87) Veröffentlichungstag der PCT-Anmeldung: **29.01.2004**
(97) Erstveröffentlichung durch das EPA: **22.06.2005**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **07.02.2007**
(47) Veröffentlichungstag im Patentblatt: **22.11.2007**

(51) Int Cl.[8]: ***G06F 1/00*** (2006.01)

(30) Unionspriorität:
202517 **23.07.2002** **US**

(84) Benannte Vertragsstaaten:
AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LI, LU, MC, NL, PT, RO, SE, SI, SK, TR

(73) Patentinhaber:
International Business Machines Corp., Armonk, N.Y., US

(74) Vertreter:
Duscher, R., Dipl.-Phys. Dr.rer.nat., Pat.-Ass., 70176 Stuttgart

(72) Erfinder:
ARNOLD, William, Carlisle, Mahopac, NY, US; CHESS, David Michael, Mohegan Lake, NY, US; MORAR, John Frederick, Mahopac, NY, US; SEGAL, Alla, Mount Kisco, NY, US; WHALLEY, Ian Nicholas, Pawling, NY, US; WHITE, Steve Richard, New York, NY, US

(54) Bezeichnung: **METHODE UND VORRICHTUNG ZUM BESTIMMEN VON POTENZIELLEM WURM-ÄHNLICHEN VERHALTEN EINES PROGRAMMES**

Beschreibung

GEBIET DER ERFINDUNG

[0001] Diese Erfindung betrifft im Allgemeinen Verfahren und Vorrichtungen zum Analysieren unerwünschter Softwareeinheiten, die zum Beispiel unter der Bezeichnung „Wurm" bekannt sind, und im Besonderen Verfahren und Vorrichtungen zum automatischen Ermitteln von potenziell wurmähnlichem Verhalten in einem Softwareprogramm.

HINTERGRUND DER ERFINDUNG

[0002] Ein Computervirus kann als ein sich selbst kopierendes Programm oder eine Softwareroutine definiert werden, die sich ohne menschliche Einwirkung in möglicherweise veränderter Form auf einem Computer verbreitet. Ein Computerwurm kann als ein Programm definiert werden, das sich im Verborgenen als Kopie zwischen Computern in einem Computernetz verbreiten kann und sich zum Kopieren des oder der Netzwerkdienste bedient.

[0003] Auf dem Gebiet der automatischen Erkennung und Analyse von Computerviren ist es oft notwendig vorherzusagen, was für ein Verhalten ein Programm zeigen wird, sodass das Programm in der für das Programm am besten geeigneten Umgebung kopiert und analysiert werden kann.

[0004] Zum Erkennen von potenziell wichtigen Verhaltensweisen (zum Beispiel wurmähnlichem Verhalten) kann Software dynamisch analysiert werden. Ein solches Verhalten kann nur sichtbar gemacht werden, wenn die Software in einer Umgebung ausgeführt wird, in der die Software wirklich oder scheinbar Zugriff auf ein Unternehmensnetz und/oder das globale Internet hat. Die Software kann in einer realen oder einer emulierten Netzumgebung ausgeführt werden, die eine Überwachungs- und eine Emulationskomponente beinhaltet. Die Überwachungskomponente dient zum Erfassen und/oder Aufzeichnen der durch die Software und/oder anderen Komponenten des Systems gezeigten Verhaltensweisen, und die Emulationskomponente vermittelt der analysierten Software den Eindruck, dass sie bei der Ausführung auf ein Unternehmensnetz und/oder das globale Internet zugreifen kann. Die analysierte Software wird streng auf die Umgebung des Analysenetzes begrenzt und kann auf keinen Fall Daten von einem Unternehmensnetz oder vom globalen Internet lesen oder dessen Daten verändern.

[0005] Es wäre wünschenswert, eine Fähigkeit zur Beschreibung der Identität von Computerwürmern außerhalb einer solchen Umgebung bereitzustellen. Zwar kann die Verwendung einer solchen Umgebung zum Kopieren sowohl von Computersoftwareviren als auch -würmern genutzt werden, jedoch kann sich dies als wenig brauchbar erweisen, da die Wurmkopierumgebung vom Vorliegen eines realen oder eines emulierten Netzes ausgeht, dessen Realisierung in der Praxis aufwändig sein kann.

[0006] Durch die Fähigkeit, außerhalb der Netzwerkumgebung vorherzusagen, ob es sich bei einem Softwareexemplar um einen potenziellen Wurm handelt, kann die Anzahl der zur Wurmkopierumgebung zu sendenden Exemplare verringert und eine wesentlich höhere Wirksamkeit automatischer Kopier- und Analysesysteme erreicht werden.

[0007] In der Beschreibung von WO-02/06 928 A2 wird ein Viruserkennungsverfahren durch das Analysieren der Verhaltensmuster von Computerprogrammen dargestellt.

ZUSAMMENFASSUNG DER ERFINDUNG

[0008] Die in den Hauptansprüchen definierte vorliegende Erfindung stellt Verfahren und Vorrichtungen zur automatischen Ermittlung des Verhaltensprofils eines Programms bereit, bei dem wurmähnliche Eigenschaften vermutet werden. Gemäß einem ersten Aspekt beinhaltet das Verfahren das Ermitteln eines Verhaltensprofils des Programms in einer Umgebung, welche den Betrieb eines Netzes nicht emuliert; das Vergleichen des ermittelten Verhaltensprofils mit einem für wurmähnliches Verhalten typischen Profil; und das Liefern eines Hinweises auf potenziell wurmähnliches Verhalten auf der Grundlage des Vergleichs. Der Schritt des Ermittelns des Verhaltensprofils umfasst folgende Schritte: Ausführen des Programms in mindestens einer bekannten nicht netzbezogenen Umgebung; Verwenden eines automatisierten Verfahrens zur Prüfung der Umgebung und zur Ermittlung, welche Änderungen gegebenenfalls in der Umgebung eingetreten sind; und Aufzeichnen aller ermittelten Änderungen in Form des Verhaltensprofils. Auf diese Weise wird ein Protokoll des beobachteten Verhaltens analysiert, um zu ermitteln, ob das Verhalten auf wurmähnliche Eigenschaften des Programms hinweist. Die nicht netzbezogene Umgebung kann dem Programm die Existenz eines Netzes vorspiegeln, ohne den Betrieb des Netzes tatsächlich zu emulieren.

KURZBESCHREIBUNG DER ZEICHNUNGEN

[0009] Im Folgenden werden lediglich beispielhaft bevorzugte Ausführungsarten der vorliegenden Erfindung unter Bezug auf die beiliegenden Zeichnungen beschrieben, wobei:

[0010] Fig. 1 ein Blockschaltbildbild eines Datenverarbeitungssystems in einer bevorzugten Ausführungsart der vorliegenden Erfindung ist;

[0011] Fig. 2 ein Schaubild eines Steuerprogramms

in einer bevorzugten Ausführungsart der vorliegenden Erfindung ist;

[0012] Fig. 3 ein logischer Ablaufplan ist, der die Arbeitsweise der Ressourcenanalysekomponente von Fig. 2 in einer bevorzugten Ausführungsart der vorliegenden Erfindung veranschaulicht;

[0013] Fig. 4A und Fig. 4B, die gemeinsam als **Fig.** 4 bezeichnet werden, einen logischen Ablaufplan zeigen, der die Arbeitsweise der Kopiereinheit (Replikator) von Fig. 2 veranschaulicht, und ein Ablaufdiagramm, das die Arbeitsweise einer Analysekomponente für Verhaltensmuster von Fig. 2 in einer bevorzugten Ausführungsart der vorliegenden Erfindung veranschaulicht.

DETAILLIERTE BESCHREIBUNG BEVORZUGTER AUSFÜHRUNGSARTEN DER ERFINDUNG

[0014] Das hier beschriebene Verfahren beruht auf den Eigenschaften eines Wurmprogramms, das in der Lage ist, sich auf andere Computer auszubreiten. Zu diesen Eigenschaften gehören unter anderem eine oder mehrere der folgenden Eigenschaften: (a) das Verwenden von Schnittstellen für die Programmierung von Anwendungsprogrammen (API) von dynamischen Bibliotheken (DLL), um elektronische Nachrichten an eine andere Maschine zu senden; (b) das Automatisieren üblicher Mailprogramme wie beispielsweise, aber nicht ausschließlich, Microsoft Outlook™ und Outlook Express™; (c) das Verwenden von Adressbüchern, einer Systemregistrierungsdatenbank und anderer Systemressourcen zum Ermitteln der potenziellen Empfänger des Wurms und/oder des Standorts von Mailprogrammen; (d) das Überschreiben oder Ersetzen von Netz-APIs in Systembibliotheken durch den Wurmcode, der bei Ausführung in alle von der Maschine als Anhang oder als separate Nachricht versendeten Mails eingefügt würde; (e) das Versuchen, auf Ressourcen in fernen Laufwerken zuzugreifen; und (f) das Löschen von Programmen, die nach dem Neustart des Systems laufen und das Versenden des Wurmcodes an andere Maschinen unter Verwendung eines der oben beschriebenen Verfahren verursachen würden.

[0015] Ein Profil eines Wurms weist solche Eigenschaften auf, die die Verwendung der oben erwähnten Verfahren erforderlich machen würden. Ein Wurmprofil kann zum Beispiel das Importieren eines Sendeverfahrens von einer Netz-DLL oder das Ändern einer Netzressource oder den Versuch des Zugriffs auf eine Systemregistrierungsdatenbank beinhalten, um Informationen zu erhalten, welche ein installiertes elektronisches Mailprogramm beschreiben.

[0016] Die Ermittlung des Verhaltensprofils eines verdächtigen Programms (d.h., bei dem wurmähnliches Verhalten, Eigenschaften oder Attribute vermutet werden) erfolgt in zwei Stufen.

[0017] Während der ersten Stufe werden die vom Programm benötigten Ressourcen ermittelt. Beispiele für Ressourcen, die auf potenziell wurmähnliches Verhalten hinweisen, sind unter anderem, aber nicht ausschließlich: (a) die dynamischen Bibliotheken für den Netzzugriff; (b) von diesen dynamischen Bibliotheken importierte Verfahren, die auf mögliche Versuche zum Versenden elektronischer Nachrichten hinweisen; und (c) dynamische Bibliotheken und Verfahren, die auf die Automatisierung vorliegender Mailprogramme hinweisen, zum Beispiel OLE oder DDE.

[0018] Während der zweiten Stufe läuft das verdächtige Programm ein oder mehrere Male in einer kontrollierten Netzwerkumgebung, wobei, wenn möglich alle, Zugriffe auf Systemressourcen, überwacht und protokolliert werden. Zur Einengung der Ergebnisse und zur Verringerung der Anzahl falscher positiver Ergebnisse kann die nicht netzbezogene Umgebung so gestaltet werden, dass sie scheinbar über bestimmte Netzeigenschaften verfügt. Wenn ein als Wurm verdächtigtes Programm zum Beispiel versucht, auf Adressbücher zuzugreifen oder das vorliegen von elektronischen Mailprogrammen prüft, kann die Umgebung die erwarteten Informationen liefern, um eine deutlichere wurmähnliche Reaktion zu provozieren. In bestimmten Fällen kann es von Vorteil sein, im System ein elektronisches Mailprogramm zu installieren, um eine positive Reaktion des verdächtigen Programms zu ermöglichen, falls dieses auf ein solches Programm zuzugreifen versucht.

[0019] Nachdem das Programm einmal oder mehrmals gelaufen ist, wird das Verhaltensprofil des Programms erstellt, wobei sowohl die am System vorgenommenen Änderungen, beispielsweise eine Änderung einer Netz-DLL, als auch jegliche Zugriffsversuche auf bestimmte Ressourcen wie beispielsweise einen Registrierungdatenbanksschlüssel, der den Standort des Mailprogramms oder eines Adressbuches enthält, zugrunde gelegt werden.

[0020] Eine bevorzugte Ausführungsart der vorliegenden Erfindung läuft auf einem oder mehreren in Fig. 1 gezeigten Computersystemen **100**, wobei die Figur ein Blockschaltbild eines typischen Rechnersystems **100** zeigt, in dem die bevorzugte Ausführungsart der vorliegenden Erfindung realisiert werden kann. Das Computersystem **100** beinhaltet eine Computerplattform **102** mit einer Hardwareeinheit **103** und ein Softwareanalyseprogramm (SAP) **101**, das im vorliegenden Zusammenhang auch als Steuerprogramm **101** bezeichnet wird und die im Folgenden beschriebenen Verfahren ausführt. Das SAP **101** läuft auf der Computerplattform **102** und der Hardwareeinheit **103**. Die Hardwareeinheit **103** beinhaltet normalerweise eine oder mehrere Zentraleinheiten

(Central Processing Units, CPUs) **104**, einen Speicher **105**, der einen Arbeitsspeicher (Random Access Memory, RAM) beinhalten kann, und eine E/A-Schnittstelle **106**. Auf der Plattform **102** können sich Anweisungen in Form von Mikrocode befinden, zum Beispiel ein reduzierter Befehlssatz. Mit der Computerplattform **102** können verschiedene Peripherieeinheiten **130** verbunden sein. Üblicherweise gehören zu den Peripherieeinheiten **130** ein Bildschirm **109**, eine externe Datenspeichereinheit (z.B. ein Band- oder Plattenspeicher) **110**, in welchen die durch die bevorzugte Ausführungsart benutzten Daten gespeichert sind und wo sich ein Ressourcenprofil **205** des Wurms (siehe Fig. 2) befindet, sowie eine Druckereinheit **111**. Ferner kann das System eine Verbindung **112** zum verbinden des Systems **100** mit einem oder mehreren ähnlichen Computersystemen beinhalten, die einfach als Kasten **113** dargestellt sind. Die Verbindung **112** dient zur Übertragung digitaler Informationen zwischen den Computern **100** und **113**. Die Verbindung **112** ermöglicht auch den Zugriff auf das globale Internet **113a**. Ein Betriebssystem (Operating System, OS) **114** steuert die Funktion der verschiedenen Komponenten des Computersystems **100** und ist für die Verwaltung verschiedener Objekte und Dateien sowie für die Aufzeichnung bestimmter diesbezüglicher Informationen zuständig, z.B. Tag und Uhrzeit der letzten Änderung, Dateilänge usw. Zum OS **114** gehören üblicherweise eine Registrierungsdatenbank **114A**, deren Verwendung im Folgenden erörtert wird, und Systeminitialisierungsdateien (SYS_INIT_DATEIEN) und andere Dateien, z.B. DLLs **114B**. Auf das OS **114** ist eine Softwaretoolschicht **116** aufgesetzt, die zum Beispiel Compiler, Interpreter und andere Softwaretools enthält. Die Interpreter, Compiler und die anderen Tools in der Schicht **116** laufen oberhalb des Betriebssystems **114** und ermöglichen die Ausführung von Programmen, die in der Technik bekannte Verfahren verwenden.

[0021] Ein geeignetes und nicht als Einschränkung anzusehendes Beispiel eines Computersystems **100** ist die IBM IntelliStation™ (Warenzeichen der International Business Corporation). Ein Beispiel einer geeigneten CPU ist ein Pentium™ III-Prozessor (Warenzeichen der Intel Corporation); Beispiele für Betriebssystemen sind Microsoft Windows™ 2000 (Warenzeichen von Microsoft Corporation) und eine Redhat-Version von GNU/Linux; Beispiele eines Interpreters und eines Compilers sind ein Perl-Interpreter und ein C++-Compiler. Der Fachmann kann sich vorstellen, dass die oben erwähnten Rechnersysteme, Prozessoren, Betriebssysteme und Tools durch andere Beispiele ersetzt werden können.

[0022] Das SAP oder Steuerprogramm **101** dient gemäß den Lehren der vorliegenden Erfindung zum Ermitteln eines wurmähnlichen Verhaltens, das sich auf der Ausführung eines verdächtigen schädlichen Programms oder einem unerwünschten Softwareexemplar ergibt, das hier allgemein als Probe **115** bezeichnet wird.

[0023] Eine gegenwärtig bevorzugte, aber nicht als Einschränkung zu verstehende, Ausführungsart des SAP oder Steuerprogramms **101** verwendet zur Ausführung eines Steuerteilsystems oder einer Steuereinheit **200** und eines Kopierteilsystems oder einer Kopiereinheit **201**, die beide in Fig. 2 dargestellt sind, einen oder mehrere Computer. Ein Analysator **202** für wurmähnliches Verhalten, der Bestandteil der Steuereinheit **200** ist, verwendet einige der derzeit vorhandenen Tools **116**, um die vom verdächtigen Programm oder der Probe **115** verwendeten DLLs und Importe zu ermitteln.

[0024] Fig. 2 ist ein detailliertes Schaubild einer Steuereinheit **200** und einer Kopiereinheit **201** bei der bevorzugten Ausführungsart der vorliegenden Erfindung und zeigt einen Analysator **202** für wurmähnliches Verhalten und die Umgebung, in welcher die bevorzugte Ausführungsart läuft. Die Umgebung beinhaltet mehrere Rechnersysteme, von denen eines eine Steuereinheit **200** und ein anderes eine Kopiereinheit **201** ist.

[0025] Anzumerken ist, dass die Einheit hier zwar als Kopiereinheit bezeichnet wird, dass aber ihre Hauptfunktion nicht im Kopieren eines Wurms, d.h. im Erzeugen eines weiteren Exemplars des Wurms, besteht. Vielmehr besteht ihre Hauptaufgabe in der Schaffung einer Systemumgebung, genauer gesagt einer emulierten Systemumgebung, in der die Probe **115** einmal oder mehrmals auf eine oder mehrere Arten ausgeführt wird, um das Verhalten der Probe **115** in Bezug auf Zustandsänderungen der Systemumgebung zu ermitteln sowie einen Datensatz oder ein Protokoll der Aktivitäten der Probe **115** zu erhalten, wenn diese in der (emulierten) Systemumgebung aktiv ist. Alle Änderungen des Systemzustands und das Protokoll der Aktivitäten der ausgeführten Probe **115** werden mit den Zustandsänderungen und Aktivitäten verglichen, die für ein wurmähnliches Verhalten bekannt sind, sodass bei Übereinstimmung davon ausgegangen wird, dass die Probe **115** wurmähnliche Eigenschaften aufweist. Erst zu diesem Zeitpunkt kann es wünschenswert sein, eine Kopie des verdächtigen Wurm herzustellen, um weitere Exemplare zur Analyse und zur anschließenden Erkennung zur Verfügung zu stellen.

[0026] Das Ausführen der Probe **115** auf verschiedene „Arten" bedeutet im Rahmen der vorliegenden Erfindung, dass das Probeprogramm einmal oder mehrmals über verschiedene System-APIs (z.B. System und/oder CreateProcess) läuft und auch eine grafische Benutzerschnittstelle (GUI) ausführt, falls das Programm eine GUI aufweist. Das Ausführen der Probe **115** auf mehrere Weisen kann auch dadurch

erfolgen, dass das Probeprogramm läuft, das System neu gestartet wird und das Programm noch einmal läuft. Diese Verfahren sind bezüglich der „Arten", in denen die Probe **115** ausgeführt werden kann, nicht als erschöpfend anzusehen.

[0027] Der Analysator **202** für wurmähnliches Verhalten beinhaltet einen Ressourcenanalysator **203**, der hier auch als statischer Analysator oder als statische Ermittlungseinheit bezeichnet wird, und einen Analysator **204** für Verhaltensmuster, der hier auch als dynamischer Analysator oder als dynamische Ermittlungseinheit bezeichnet wird. Der Analysator **202** für Verhaltensmuster verwendet Tools **206** und **207**, welche eine Liste von der Probe **115** benötigter dynamischer Bibliotheken **114B** bzw. die aus den dynamischen Bibliotheken **114B** importierten Verfahren ermitteln. Ein Beispiel eines Tool **206**, **207**, das zum Ermitteln verwendet werden kann, welche dynamischen Bibliotheken ein Programm benötigt, ist unter der Bezeichnung Microsoft DEPENDS.EXE bekannt und wird in einem Artikel mit dem Titel „Under the Hood" von Matt Pietrik in der Ausgabe des Microsoft Systems Journal vom Februar 1997 beschrieben, die über das Microsoft Developer Network erhältlich ist. Ein Beispiel eines Tool **206**, **207**, das zum Ermitteln der Importe des Probeprogramms **115** verwendet werden kann, ist unter der Bezeichnung DUMPBIN.EXE bekannt, das Bestandteil von Microsoft Developer Studio™, Version 6.0, ist. Der Fachmann kann andere Tools zum Ausführen derselben oder ähnlicher Funktionen beschaffen oder schreiben.

[0028] Der Ressourcenanalysator **203** (statische Ermittlung) nutzt diese Tools zum Erzeugen eines Profils der durch das verdächtige Programm oder die Probe **115** genutzten Ressourcen und vergleicht die Ergebnisse mit dem Inhalt des Ressourcenprofils **205** des Wurms. Ein typisches Ressourcenprofil **205** eines Wurms kann unter anderem beispielsweise die Netzwerk-DLLs **114B** WSOCK32.DLL, INETMIB1.DLL, WINSOCK.DLL, MSWSOCK.DLL, WININET.DLL, MAPI32.DLL, MAPI.DLL und WS2_32.DLL sowie eine DLL zum Anzeigen der OLE-Automatisierung wie beispielsweise die OLE32.DLL sowie die Liste der von diesen dynamischen Bibliotheken importieren Verfahren beinhalten. Zu diesen importierten Verfahren können unter anderem die aus der WSOCK32.DLL importierten Verfahren „send" (Senden), „send to" (Senden an) und WSAsend, die aus der OLE32.DLL importierten Verfahren CoCreateInstance und CoCreateInstanceEx oder das aus der USER32.DLL importierte Verfahren DDEConnect gehören.

[0029] Der Analysator **204** für Verhaltensmuster (dynamische Ermittlung) erzeugt das Verhaltensprofil eines verdächtigen Programms unter Verwendung der Ergebnisse des Durchlaufs der Probe **115** durch die Kopiereinheit **201** und vergleicht diese Ergebnisse mit einem Verhaltensprofil **208** des Wurms. Ein typisches Verhaltensprofil **208** des Wurms beinhaltet eine Liste von Systemänderungen und/oder Zugriffsversuchen auf Dateien und Registrierungsdatenbank, die für ein wurmähnliches Verhalten kennzeichnend sind. Die Liste mit dem Verhaltensprofil **208** des Wurms kann unter anderem die folgenden Elemente beinhalten: (a) das Ändern einer oder mehreren Netzwerk-DLLs, jedoch nicht der nicht netzbezogenen DLLs; (b) das Erzeugen einer oder mehrerer Dateien mit der Erweiterung VBS; (c) das Erzeug neuer Dateien und entsprechende Änderungen an den bzw. das Erzeugen der Systeminitialisierungsdateien **114B**, die ein Ersetzen von beliebigen Netz-DLLs durch eine neue Datei bewirken würden. Ein Beispiel für dieses letztere Szenario ist das Erzeugen der Datei wininit.ini im Windowsverzeichnis in einem Windowssystem, wobei die erzeugte Datei wininit.ini Befehle wie „WSOCK32.DLL=SOME.FILE" enthält und SOME.FILE eine durch das Programm erzeugte neue Datei ist. Die Liste mit dem Verhaltensprofil **208** des Wurms kann ferner (d) ein Protokoll der Zugriffsversuche auf Adressbücher und/oder oder Registrierungsdatenbankschlüssel beinhalten, die dem Standort eines elektronischen Mailprogramms entsprechen.

[0030] Sowohl das Ressourcenprofil **205** als auch das Verhaltensprofil **208** des Wurms sind von Natur aus statisch und werden vorzugsweise erzeugt, bevor die Probe **115** auf der Kopiereinheit **201** läuft.

[0031] Die Kopiereinheit **201** wird durch die Steuereinheit **200** aufgerufen, bevor die oben erörterte Analyse des Verhaltensmusters in Gang gesetzt wird. Die Kopiereinheit **201** beinhaltet eine Kopiersteuerung (Replication Controller, RC) **209**, Verhaltensüberwachungseinheiten **210** und wahlweise eine Simulationseinheit **211** für Netzwerkverhalten.

[0032] Die Simulationseinheit **211** für Netzwerkverhalten erzeugt gemeinsam mit den Verhaltensüberwachungseinheiten **210** ein scheinbares netzähnliches Verhalten, um bestimmte wurmähnliche Verhaltensweisen der Probe **115** aufzudecken. Zum Beispiel bietet die Simulationseinheit **211** für Netzwerkverhalten der Probe **115** eine falsche Netzadresse an, zum Beispiel eine falsche IP-Adresse, wenn die Verhaltensüberwachungseinheit **210** eine Anforderung seitens der Probe **115** nach einer IP-Adresse erkennt. In diesem Fall kann die Probe **115** vor der Offenbarung des wurmähnlichen Verhaltens die lokale IP-Adresse anfordern, um zu überprüfen, ob das System über Netzfunktionen verfügt, worauf der Probe **115** eine lokale IP-Adresse angeboten werden kann, um die Probe **115** zur Offenbarung des wurmähnlichen Verhaltens zu veranlassen.

[0033] Auf ähnliche Weise kann die Umgebung, in der die Probe **115** läuft, so gestaltet werden, dass sie

die Existenz von Systemressourcen und/oder -objekten vorspiegelt, die in Wirklichkeit nicht vorhanden sind. Wenn die Probe **115** beispielsweise Informationen über eine bestimmte Datei anfordert, kann es von Vorteil sein, dass die Umgebung mit den angeforderten Informationen so antwortet, als gebe es die Datei, oder die Datei vor dem Zurückgeben zur Probe **115** erzeugt, um die Probe zur Offenbarung des wurmähnlichen Verhaltens zu veranlassen. Das heißt, eine bekannte nicht netzbezogene Umgebung kann so gestaltet werden, dass sie dem Programm nicht vorhandene lokale netzbezogene Ressourcen oder lokale netzbezogene Objekte vorspiegelt.

[0034] Fig. 3 und 4 veranschaulichen den Ablauf der Ausführung durch die Steuereinheit **200** und die Kopiereinheit **201**. In Fig. 3 wird die Probe **115** zuerst in Schritt **301** zur Steuereinheit **200** gesendet, welche die Probe **115** dann in Schritt **302** zum Ressourcenanalysator **203** weiterleitet. Der Ressourcenanalysator **203** ermittelt in Schritt **303**, auf welche dynamischen Bibliotheken **114B** die Probe **115** zugreift, und vergleicht in Schritt **304** die DLLs, auf die zugegriffen wurde, mit dem Ressourcenprofil **205** des Wurms.

[0035] Wenn die DLL-Nutzung mit dem Ressourcenprofil **205** des Wurms übereinstimmt, werden in Schritt **305** die aus diesen DLLs importierten Verfahren ermittelt. Wenn diese Verfahren in Schritt **306** mit denen im Ressourcenprofil **205** des Wurms übereinstimmen, wird die Probe **115** als potenzieller Wurm klassifiziert. Wenn weder die DLL-Nutzung noch die importierten Verfahren mit dem Ressourcenprofil **205** des Wurms übereinstimmen, entsprechend dem Ergebnis NEIN in einem der Schritte **304** oder **306**, wird die Probe **115** zur Kopiereinheit **201** weitergeleitet (Fig. 4A), um sie dort zu kopieren und anschließend ihr Verhaltensmuster zu ermitteln.

[0036] **Fig.** 4 veranschaulicht den Ablauf der Ausführung in der Kopiereinheit **201** (Fig. 4A) und im Verhaltensmusteranalysator **204** (Fig. 4B). Nach dem Kopieren wird die Umgebung in Schritt **401** initialisiert, die Probe in Schritt **402** zur Kopiereinheit **201** gesendet und in Schritt **403** ausgeführt. Dann geht der Prozess weiter zum Verhaltensmusteranalysator **204** (Fig. 4B), der in Schritt **404** die Änderungen des Systems untersucht und in Schritt **405** die ermittelten Änderungen mit denen im Verhaltensprofil des Wurms vergleicht. Bei Übereinstimmung wird die Probe **115** als potenzieller Wurm eingestuft, ansonsten analysiert der Verhaltensmusteranalysator **204** in Schritt **406** die durch die Verhaltensüberwachungseinheiten **210** mitgeteilten Aktivitäten und versucht in Schritt **407**, das mitgeteilte Probenverhalten mit den Aktivitätsmustern (wurmähnliche Verhaltensmuster) zu vergleichen, die im Verhaltensprofil **208** des Wurms aufgeführt sind.

[0037] Wenn entweder die in Schritt **404** analysierten Systemänderungen oder die in Schritt **406** durch die Verhaltensüberwachungseinheiten mitgeteilten Aktivitäten auch nur eines der im Verhaltensprofil **208** des Wurms aufgeführten Muster aufweisen, wird die Probe als potenzieller Wurm klassifiziert, während die Probe **115** ansonsten so klassifiziert wird, dass sie kein Wurm ist. Wenn die Probe **115A** in Schritt **407** als Wurm klassifiziert wird, kann sie zur weiteren Untersuchung zu einem Wurmkopier- und -analysesystem weitergeleitet werden.

[0038] Das oben beschriebene Verfahren kann auf einem computerlesbaren Medium, zum Beispiel auf der Speicherplatte **110**, installiert werden, um ein Verfahren zur automatischen Ermittlung des Verhaltensprofils des Probenprogramms **115** auszuführen, bei dem wurmähnliche Eigenschaften vermutet werden. Die Ausführung des Computerprogramms veranlasst den Computer **100**, **200** zuerst, die für das Probenprogramm **115** erforderlichen Systemressourcen des Computers zu analysieren und, wenn die erforderlichen Ressourcen noch keinen Hinweis auf wurmähnliche Eigenschaften des Probenprogramms **115** geben, das Programm in einer kontrollierten nicht netzbezogenen Umgebung auf dem Computer **100**, **200**, **201** laufen zu lassen und dabei die Zugriffe auf Systemressourcen zu überwachen und zu protokollieren, um das Verhalten des Programms in der nicht netzbezogenen Umgebung zu ermitteln. Im Laufe der weiteren Ausführung kann das Computerprogramm den Computer veranlassen, dem Probenprogramm das Vorhandensein eines Netzes vorzuspiegeln, ohne die Funktion des Netzes zu emulieren.

[0039] Bei der vorhergehenden Beschreibung ist zu beachten, dass die Kopiereinheit **201** (dynamische Ermittlung) nur tätig wird, wenn die Systemressourcen-Anforderungen der Probe noch keine potenziellen Wurmeigenschaften erkennen lassen (statische Ermittlung). Dies ist bei der gegenwärtig bevorzugten Ausführungsart der Fall, da der statische Ermittlungsprozess normalerweise weniger rechenintensiv, aber wesentlich schneller als der dynamische Ermittlungsprozess ist. Jedoch können auch beide Prozesse oder Teilsysteme ausgeführt werden, wenn durch den ersten Prozess eine wurmähnliche Eigenschaft angezeigt wird und der zweite Prozess oder das zweite Teilsystem zur Überprüfung des Ergebnisses des ersten Prozesses dient. Ferner ist anzumerken, dass es in bestimmten Fällen wünschenswert sein kann, dass sowohl der statische als auch der dynamische Ermittlungsprozess bezüglich der wurmähnlichen Eigenschaften eines bestimmten Probe **115** zum gleichen Ergebnis kommen, da es zu falschen negativen Ergebnissen kommen kann. Deshalb wird gegenwärtig bevorzugt, dass einer der beiden Prozesse, der statische oder der dynamische Prozess, dafür zuständig ist, eine bestimmte Probe als wurmähnlich zu deklarieren, sodass das System zur Verarbeitung der Probe als potenzieller Virus übergehen

kann, wenn kein wurmähnliches Verhalten angezeigt wird.

[0040] Die Probe **115** kann normalerweise wie oben erwähnt mittels der durch den Ressourcenanalysator **203** (außerhalb der emulierten Umgebung) durchgeführten statischen Ermittlung schneller verarbeitet werden als mittels der durch den Verhaltensmusteranalysator **204** durchgeführten dynamischen Ermittlung.

Patentansprüche

1. Verfahren zur automatischen Ermittlung von potenziell wurmähnlichem Verhalten eines Programms, wobei das Verfahren Folgendes umfasst:
Ermitteln eines Verhaltensprofils (**303**) des Programms in einer Umgebung, welche den Betrieb eines Netzes nicht emuliert, wobei die Umgebung über eine Fähigkeit verfügt, sich so zu präsentieren, als verfüge sie über netzbezogene Fähigkeiten (**201**);
Vergleichen des ermittelten Verhaltensprofils mit einem Profil, das auf ein wurmähnliches Verhalten hinweist (**304**); und
Bereitstellen eines Hinweises auf potenziell wurmähnliches Verhalten auf der Grundlage des Vergleichsergebnisses (**306**),
wobei der Schritt der Ermittlung des Verhaltensprofils Folgendes umfasst:
Ausführen des Programms in mindestens einer bekannten nicht netzbezogenen Umgebung;
Verwenden eines automatisierten Verfahrens zur Prüfung der Umgebung und zur Ermittlung, welche Änderungen gegebenenfalls in der Umgebung eingetreten sind; und
Aufzeichnen aller ermittelten Änderungen in Form des Verhaltensprofils.

2. Verfahren nach Anspruch 1, bei dem als Reaktion auf das Programm zur Ermittlung, ob die Umgebung über netzbezogene Fähigkeiten verfügt, dem Programm eine Netzadresse zur Verfügung gestellt wird, über welche das Programm wurmähnliches Verhalten anzeigen soll.

3. Verfahren nach Anspruch 1, bei dem die bekannte nicht netzbezogene Umgebung nicht vorhandene lokale netzbezogene Ressourcen und/oder netzbezogene Objekte für das Programm aufweist.

4. Verfahren nach Anspruch 1, bei dem als Reaktion auf das Programm zur Ermittlung von Informationen über eine Datei so reagiert wird, als stelle die Datei einen Auslöser für das Programm dar, um wurmähnliches Verhalten anzuzeigen.

5. Verfahren nach Anspruch 1, bei dem als Reaktion auf das Programm zur Ermittlung von Informationen über eine Datei diese Datei erzeugt wird, bevor die Datei an das Programm als Auslöser für das Programm zurückgegeben wird, um wurmähnliches Verhalten anzuzeigen.

6. Verfahren nach Anspruch 1, bei dem als Reaktion auf das Programm zur Ermittlung von Informationen über ein elektronisches Mailprogramm diese Informationen an das Programm als Auslöser für das Programm zurückgegeben werden, um wurmähnliches Verhalten anzuzeigen.

7. Verfahren nach Anspruch 1, bei dem als Reaktion auf das Programm zur Ermittlung von Informationen über ein elektronisches Mailadressbuch diese Informationen an das Programm als Auslöser für das Programm zurückgegeben werden, wurmähnliches Verhalten anzuzeigen.

8. Datenverarbeitungssystem, das mindestens einen Computer zum Ausführen eines gespeicherten Programms zum Durchführen eines automatisierten Ermittelns von potenziell wurmähnlichem Verhalten eines Programms umfasst, wobei das Datenverarbeitungssystem Folgendes umfasst:
ein Mittel zur
Ermitteln eines Verhaltensprofils (**303**) des Programms in einer Umgebung, welche den Betrieb eines Netzes nicht emuliert, wobei die Umgebung über eine Fähigkeit verfügt, sich so zu präsentieren, als verfüge sie über netzbezogene Fähigkeiten (**201**);
ein Mittel zum Vergleichen des ermittelten Verhaltensprofils mit einem Profil, das auf ein wurmähnliches Verhalten hinweist (**304**); und
ein Mittel zum Bereitstellen eines Hinweises auf potenziell wurmähnliches Verhalten auf der Grundlage des Vergleichsergebnisses (**306**),
wobei das Mittel zur Ermittlung des Verhaltensprofils Folgendes umfasst:
ein Mittel zum Ausführen des Programms in mindestens einer bekannten nicht netzbezogenen Umgebung;
ein Mittel zum Verwenden eines automatisierten Verfahrens zur Prüfung der Umgebung und zur Ermittlung, welche Änderungen gegebenenfalls in der Umgebung eingetreten sind; und
ein Mittel zum Aufzeichnen aller ermittelten Änderungen im Verhaltensprofil.

9. Computerprogrammprodukt, das Anweisungen umfasst, die bei Ausführung auf einem Datenverarbeitungssystem mit einer nichtflüchtigen Speichereinheit das System zum Ausführen eines Verfahrens nach einem der Ansprüche 1 bis 7 veranlasst.

Es folgen 4 Blatt Zeichnungen

Anhängende Zeichnungen

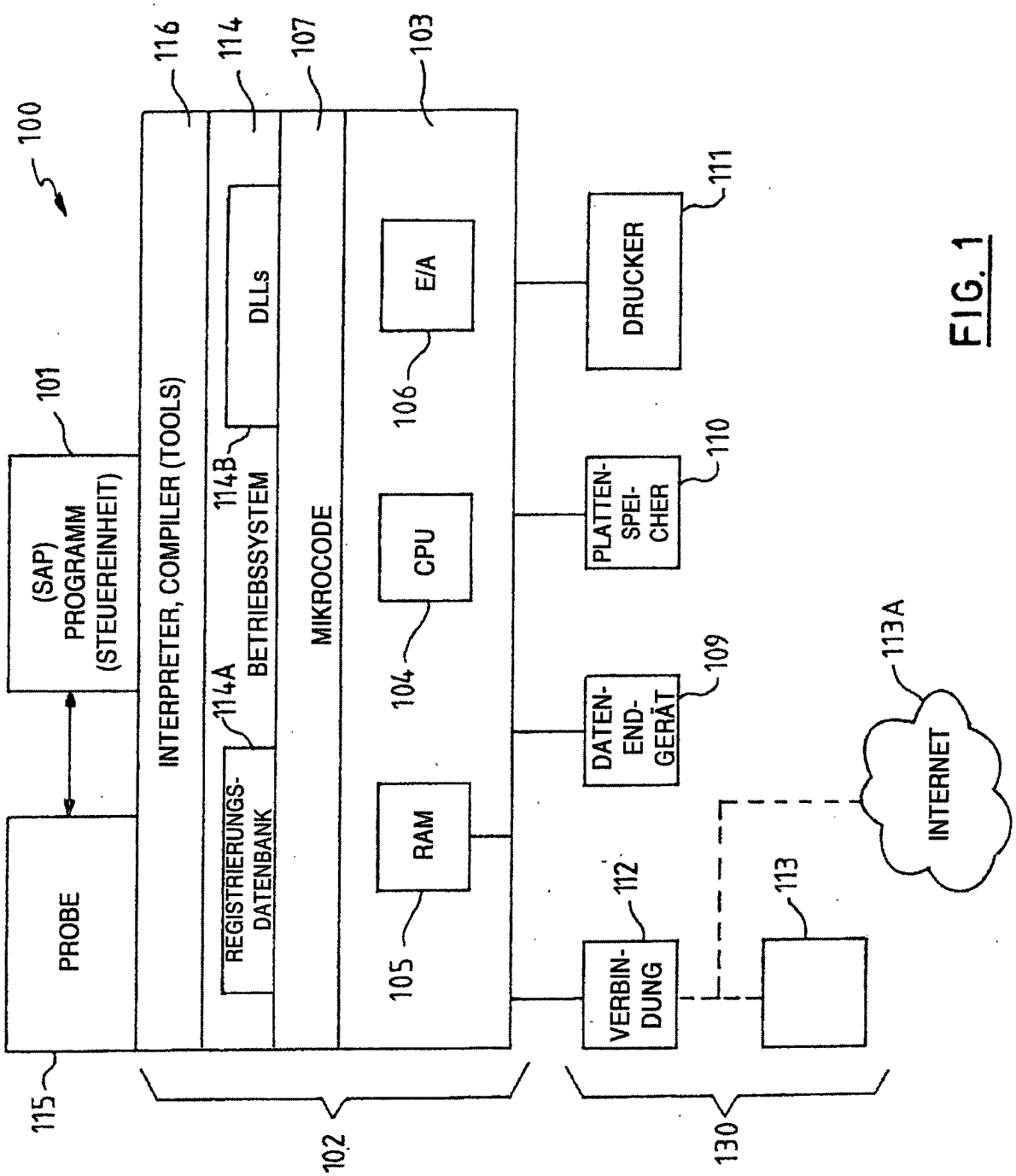

FIG. 1

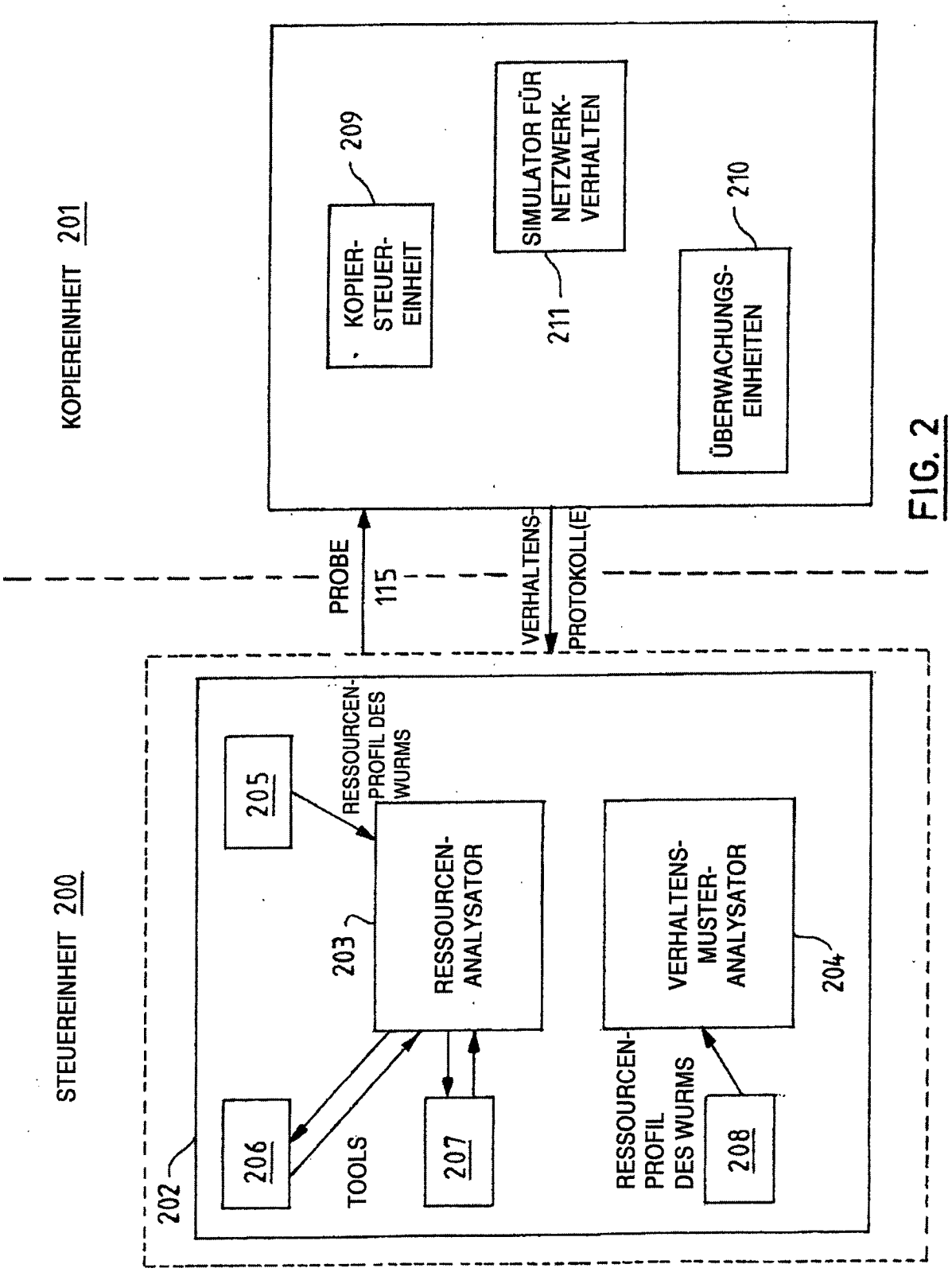
KOPIEREINHEIT 201
KOPIER-STEUER-EINHEIT
209
SIMULATOR FÜR NETZWERK-VERHALTEN
211
ÜBERWACHUNGS-EINHEITEN
210
PROBE
115
VERHALTENS-PROTOKOLL(E)
STEUEREINHEIT 200
205
RESSOURCEN-PROFIL DES WURMS
RESSOURCEN-ANALYSATOR
203
VERHALTENS-MUSTER-ANALYSATOR
204
206
TOOLS
207
RESSOURCEN-PROFIL DES WURMS
208
202

FIG. 2

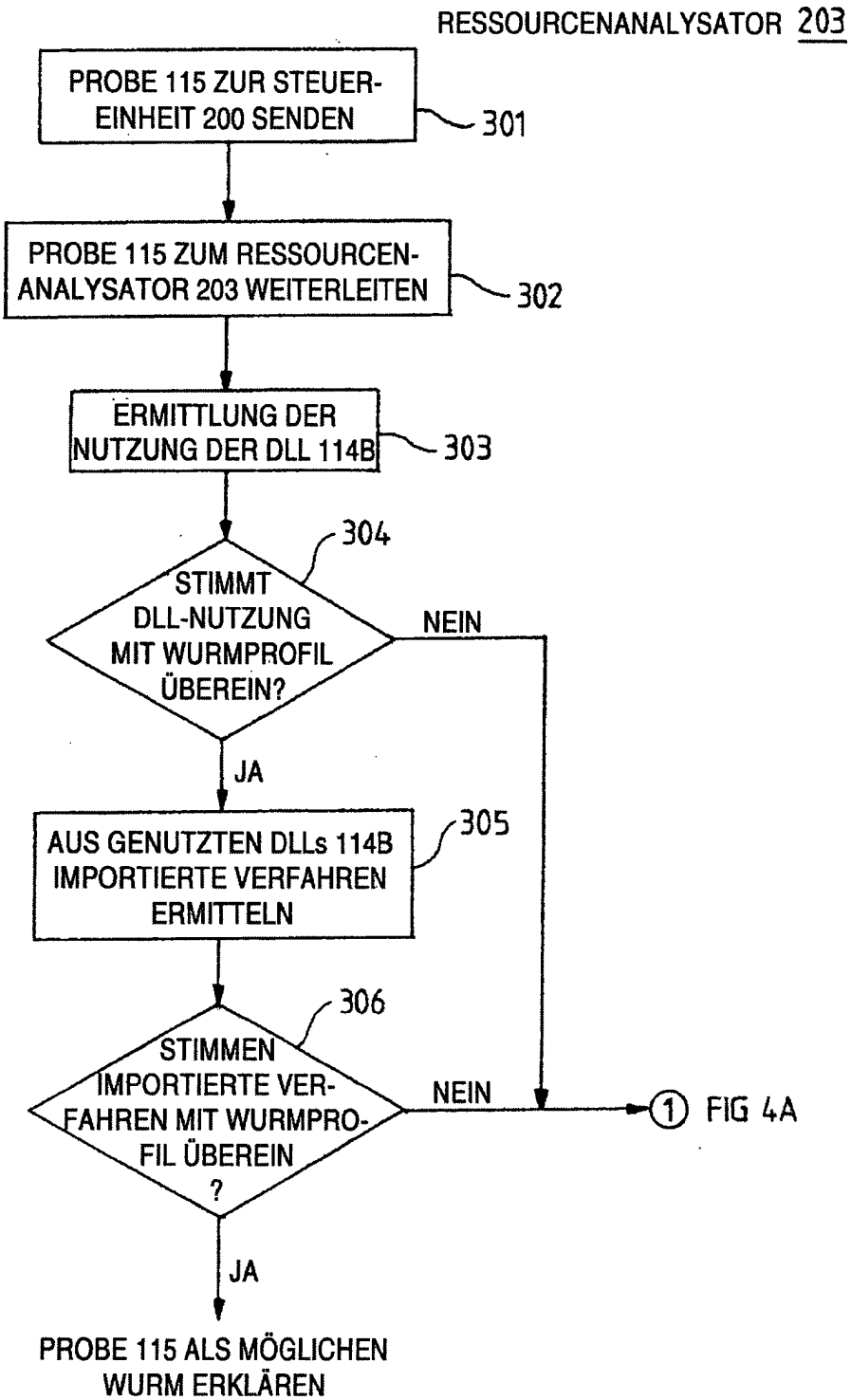
RESSOURCENANALYSATOR 203
PROBE 115 ZUR STEUER-EINHEIT 200 SENDEN
301
PROBE 115 ZUM RESSOURCEN-ANALYSATOR 203 WEITERLEITEN
302
ERMITTLUNG DER NUTZUNG DER DLL 114B
303
304
STIMMT DLL-NUTZUNG MIT WURMPROFIL ÜBEREIN?
NEIN
JA
AUS GENUTZTEN DLLs 114B IMPORTIERTE VERFAHREN ERMITTELN
305
306
STIMMEN IMPORTIERTE VER-FAHREN MIT WURMPRO-FIL ÜBEREIN ?
NEIN
① FIG 4A
JA
PROBE 115 ALS MÖGLICHEN WURM ERKLÄREN

FIG. 3

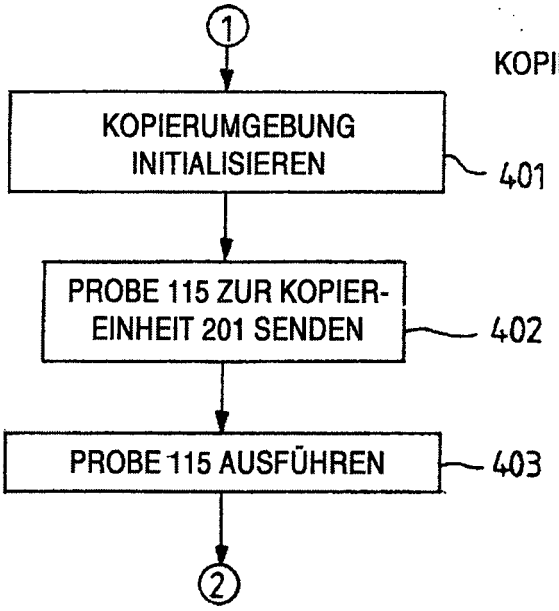
1
KOPIEREINHEIT 201
KOPIERUMGEBUNG INITIALISIEREN
401
PROBE 115 ZUR KOPIER-EINHEIT 201 SENDEN
402
PROBE 115 AUSFÜHREN
403
2

FIG. 4A

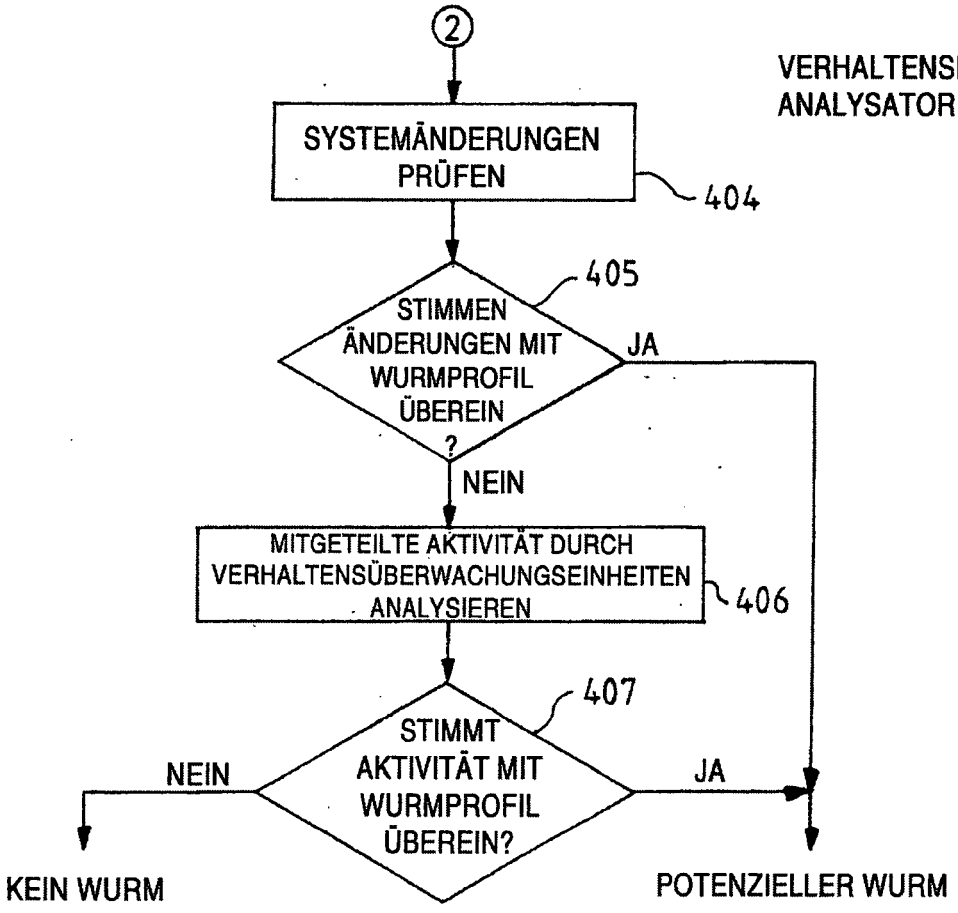
2
VERHALTENSMUSTER-ANALYSATOR 204
SYSTEMÄNDERUNGEN PRÜFEN
404
405
STIMMEN ÄNDERUNGEN MIT WURMPROFIL ÜBEREIN ?
JA
NEIN
MITGETEILTE AKTIVITÄT DURCH VERHALTENSÜBERWACHUNGSEINHEITEN ANALYSIEREN
406
407
STIMMT AKTIVITÄT MIT WURMPROFIL ÜBEREIN?
NEIN
JA
KEIN WURM
POTENZIELLER WURM

FIG. 4B