(19) **Bundesrepublik Deutschland**
Deutsches Patent- und Markenamt

(10) **DE 601 11 089 T2** 2006.05.04

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 1 119 151 B1**
(21) Deutsches Aktenzeichen: **601 11 089.7**
(96) Europäisches Aktenzeichen: **01 300 123.5**
(96) Europäischer Anmeldetag: **08.01.2001**
(97) Erstveröffentlichung durch das EPA: **25.07.2001**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **01.06.2005**
(47) Veröffentlichungstag im Patentblatt: **04.05.2006**

(51) Int Cl.[8]: ***H04L 29/06*** (2006.01)
H04L 12/24 (2006.01)
H04L 12/22 (2006.01)

(30) Unionspriorität:
483876 **18.01.2000** **US**

(84) Benannte Vertragsstaaten:
DE, FR, GB, IT

(73) Patentinhaber:
Lucent Technologies Inc., Murray Hill, N.J., US

(74) Vertreter:
derzeit kein Vertreter bestellt

(72) Erfinder:
Mayer, Alain, New York, US; Wool, Avishai, Livingston, US; Ziskind, Elisha, Princeton, US

(54) Bezeichnung: **Verfahren und Vorrichtung zum Analysieren von einer oder mehrerer Firewalls**

Beschreibung

Technisches Gebiet

[0001] Die vorliegende Erfindung betrifft allgemein Firewalls und insbesondere ein Verfahren und eine Vorrichtung zur Analyse der Sicherheitsrichtlinie einer Firewall.

Allgemeiner Stand der Technik

[0002] Netzwerk-Firewalls liefern wichtige Schutzeinrichtungen für jedes mit dem Internet verbundene Netzwerk. Firewalls sind keine einfachen Anwendungen, die „aus dem Kasten" aktiviert werden können. Eine Firewall muß konfiguriert und verwaltet werden, um eine wichtige Sicherheitsrichtlinie für die bestimmten Bedürfnisse einer gegebenen Firma oder Entität zu realisieren. Es wurde gesagt, daß der wichtigste Faktor, der sich auf die Sicherheit einer Firewall auswirkt, die Firewall-Konfiguration ist. Obwohl bei Firewalls beeindruckende technische Fortschritte erzielt wurden, gab es nur wenige oder überhaupt keine Fortschritte bei der Firewall-Konfiguration und -Verwaltung.

[0003] Eine Firewall ist ein Netzwerk-Gateway, das Pakete filtert und ein proprietäres Firmennetzwerk, wie zum Beispiel ein Intranet, von einem öffentlichen Netzwerk wie zum Beispiel dem Internet trennt. Die meisten der derzeitigen Firewalls werden mittels einer Regelbasis oder Firewall-Konfigurationsdatei konfiguriert. Im Fall einer Firewall, die ein einziges, homogenes Intranet, wie zum Beispiel das lokale Netzwerk (LAN) einer kleinen Firma schützt, instruiert eine einzige Regelbasis die Firewall, welche ankommenden Sitzungen (Pakete) sie durchlassen darf und welche blockiert werden sollten. Ähnlich spezifiziert die Regelbasis, welche abgehenden Sitzungen (Pakete) zugelassen werden. Der Firewall-Administrator muß die Firmensicherheitsrichtlinien auf hoher Ebene mit dieser Regelbasis auf niedriger Ebene implementieren.

[0004] Die Konfigurationsschnittstelle der Firewall erlaubt es in der Regel dem Sicherheitsadministrator, verschiedene Hostgruppen (Bereiche von IP-Adressen) und Dienstgruppen (Gruppen von Protokollen und entsprechenden Portnummern an den Host, die die Endpunkte bilden) zu definieren. Eine einzelne Regel enthält typischerweise eine Quelle, ein Ziel, eine Dienstgruppe und eine entsprechende Aktion. Quelle und Ziel sind Hostgruppen und die Aktion ist im allgemeinen entweder eine Anzeige, die Pakete der entsprechenden Sitzung „durchzulassen" oder „abzukoppeln".

[0005] In vielen Firewalls ist die Regelbasis reihenfolgeabhängig. Anders ausgedrückt prüft die Firewall, ob die erste Regel in der Regelbasis für eine neue Sitzung gilt. Wenn die erste Regel gilt, werden die Pakete gemäß der durch die erste Regel spezifizierten Aktion entweder durchgelassen oder abgekoppelt. Andernfalls prüft die Firewall, ob die zweite Regel gilt, und so weiter, bis eine Regel gilt. Dieses Schema führt häufig zu einer Fehlkonfiguration aufgrund redundanter Regeln in der Regelbasis, und die gewünschte Sicherheitsrichtlinie wird erst nach Umordnung eines Teils der Regeln realisiert.

[0006] Die Probleme der Administration einer Firewall sind für eine größere Firma, die möglicherweise mehr als eine einzige Firewall verwendet, sogar noch schwerwiegender. Mehrere Firewalls unterteilen die Intranets einer Firma in mehrere Zonen und die Sicherheitsrichtlinie wird typischerweise durch mehrere Regelbasen realisiert, die sich auf mehreren Gateways befinden, die die verschiedenen Zonen miteinander verbinden. Das Zusammenspiel zwischen den verschiedenen Regelbasen muß also sorgfältig untersucht werden, damit keine Sicherheitslöcher eingeführt werden. Die Komplexität des Entwurfs und der Verwaltung der Regelbasen wächst mit komplexer werdenden Intranets.

[0007] Heutzutage enthält sogar ein mäßig großes Firmenintranet mehrere Firewalls und Router, die alle dazu verwendet werden, verschiedene Aspekte der globalen Firmensicherheitsrichtlinie durchzusetzen. Die Konfiguration dieser Einrichtungen zur gleichzeitigen Arbeit ist schwierig, insbesondere wenn die Einrichtungen von verschiedenen Vertreibern hergestellt werden. Sogar das Prüfen oder Reverse Engineering einer existierenden Konfiguration ist zum Beispiel schwer, wenn ein neuer Sicherheitsadministrator übernimmt. Firewall-Konfigurationsdateien werden in Formalismen auf niedriger Ebene geschrieben, deren Lesbarkeit mit Assemblercode vergleichbar ist, und die globale Richtlinie ist über alle beteiligten Firewalls verteilt.

[0008] Zur Zeit besitzen Firewall-Administratoren keine einfache Möglichkeit zur Bestimmung der Sicherheitsrichtlinien, die für verschiedene Klassen von Maschinen oder Diensten in einer Firmenumgebung gelten. Es kann also für den Administrator schwierig oder sogar unmöglich sein, Routinefragen bezüglich der Firmensicherheitsrichtlinien zu beantworten, wie zum Beispiel ob ein oder mehrere gegebene Dienste zwischen einem

oder mehreren gegebenen Maschinen erlaubt sind. Die Bewertung der Firmensicherheitsrichtlinien kann aus mehreren Gründen schwierig sein. Erstens können Pakete mehrere Wege zwischen einer Quelle und einem Ziel aufweisen, wobei jeder Weg mehrere Filterungseinrichtungen durchquert. Um eine Anfrage zu beantworten, müßte der Administrator die Regeln aller dieser prüfen. Zusätzlich behandeln typische Vertreiberkonfigurationswerkzeuge nur eine einzige Einrichtung auf einmal, was zu einem uneinheitlichen globalen Verhalten führen kann. Wenn von verschiedenen Vertreibern hergestellte Paketfilterungseinrichtungen beteiligt sind, verschlimmert sich die Situation schnell. Darüber hinaus ist sogar das Verständnis der Richtlinie auf einer einzigen Schnittstelle einer einzigen Paketfilterungseinrichtung problematisch. Wie bereits erwähnt, sind Firewall-Konfigurationssprachen tendenziell mysteriös, befinden sich auf sehr niedriger Ebene, sind von der Regelreihenfolge abhängig und sehr vertreiberspezifisch.

[0009] Zur Zeit sind mehrere Anfälligkeitsprüfwerkzeuge kommerziell erhältlich. Zum Beispiel versucht Satan, das zum Beispiel in M. Freiss, „Protecting Networks with SATAN", O'Reilly & Associates, Inc. (1998) beschrieben wird, bekannte Anfälligkeiten in vielfach eingesetzten Protokollen und Betriebssystemen auszunutzen, von denen einige durch entsprechende Firewall-Richtlinien blockiert werden können. Auf diese Weise kann man mit Satan die Firewall-Richtlinie prüfen. Zusätzlich verbindet sich NetSonar 2.0™, das kommerziell von Cisco Systems Inc. in San Jose, CA, erhältlich ist, mit einem Firmenintranet und sondiert das Netzwerk und prüft dadurch die verwendeten Routing- und Firewall-Richtlinien.

[0010] Die zur Zeit verfügbaren Anfälligkeitsprüfwerkzeuge sind aktiv. Anders ausgedrückt, senden und empfangen sie Pakete auf dem Netzwerk. Folglich unterliegen sie mehreren Beschränkungen, die, wenn sie überwunden werden, die Nützlichkeit und Effizienz solcher Anfälligkeitsprüfwerkzeuge stark erweitern könnten. Wenn zum Beispiel das Intranet groß ist (mit vielen tausenden Maschinen), sind derzeitige Anfälligkeitsprüfwerkzeuge entweder langsam (wenn sie jede einzelne IP-Adresse im Vergleich zu jedem möglichen Port prüfen) oder statistisch (wenn sie zufällig prüfen). Sie können sicherlich nicht jede mögliche IP-Adresse im Internet prüfen.

[0011] Außerdem können zur Zeit erhältliche Anfälligkeitsprüfwerkzeuge nur eine Art von Firewall-Konfigurationsfehler erfassen: das Durchlassen von unautorisierten Paketen. Sie erfassen nicht die zweite Art von Fehler: unbeabsichtigtes Blockieren autorisierter Pakete. Diese zweite Art von Fehler wird typischerweise durch eine Strategie des „Einsetzens und Wartens auf Beschwerden" erkannt, die für die Netzwerkbenutzer störend ist und kritische Unternehmensanwendungen abschneiden kann.

[0012] Eine aktive Prüfung erfolgt immer nach dem Faktum. Das Erkennen eines Problems, nachdem die neue Richtlinie eingesetzt wurde, ist jedoch (a) gefährlich (da das Netzwerk anfällig ist, bis das Problem erkannt und eine sichere Richtlinie eingesetzt wurde), (b) kostspielig (da die Verwendung einer Sicherheitsrichtlinie in einem großen Netzwerk eine zeitaufwendige und fehleranfällige Aufgabe ist) und (c) störend für Benutzer. Ferner kann ein aktives Werkzeug nur von seinem physischen Ort in der Netzwerktopologie aus prüfen. Ein für einen Weg durch das Netzwerk, in dem der Host nicht vorkommt, auf dem das aktive Werkzeug abläuft, spezifische Problem bleibt unerkannt.

[0013] Es wird deshalb ein Firewall-Analysewerkzeug benötigt, das es einem Administrator erlaubt, eine globale Firewall-Richtlinie zu entdecken und zu prüfen. Ferner wird ein Firewall-Analysewerkzeug benötigt, das eine minimale Beschreibung der Netzwerktopologie verwendet und die verschiedenen vertreiberspezifischen Konfigurationsdateien auf niedriger Ebene direkt analysiert. Ferner wird ein Firewall-Analysewerkzeug benötigt, das durch eine Frage- und Antwort-Sitzung mit dem Benutzer in Dialog tritt, die auf einem entsprechenden Abstraktionsniveau durchgeführt wird.

[0014] Aus US-B-5 968 176 ist ein System bekannt, das die Herstellung von Sicherheit in einem Netzwerk gewährleistet und das Knoten enthält, die Sicherheitsfunktionen aufweisen, die in mehreren Protokollschichten wirken. Mehrere Netzwerkeinrichtungen, wie zum Beispiel Fernzugangsgeräte, Router, Switches, Zwischenverstärker und Netzwerkkarten, besitzen Sicherheitsfunktionen durch deren Konfiguration zur Implementierung von verteilten Firewall-Funktionen in dem Netzwerk beigetragen wird. Durch Verteilung von Firewall-Funktionalität auf viele Schichten des Netzwerks in vielfältigen Netzwerkeinrichtungen wird eine alles durchdringende Firewall implementiert. Die alles durchdringende, mehrschichtige Firewall enthält eine Richtliniendefinitionskomponente, die Richtliniendaten annimmt, die definieren, wie sich die Firewall verhalten soll. Die Richtliniendefinitionskomponente kann eine zentralisierte Komponente sein, oder eine Komponente, die über das Netzwerk verteilt ist. Die mehrschichtige Firewall enthält auch eine Ansammlung von Netzwerkeinrichtungen, mit denen die definierte Richtlinie durchgesetzt wird. Die Sicherheitsfunktionen, die in dieser Ansammlung von Netzwerkeinrichtungen über mehrere Protokollschichten hinweg wirken, werden durch die

Richtliniendefinitionskomponete so koordiniert, daß bestimmte Einrichtungen den Teil der Richtlinie durchsetzen, der ihren Teil des Netzwerks betrifft.

[0015] Hinrichs S: „Policy-based management: briding the gap" COMPUTER SECURITY APPLICATIONS CONFERENCE, 1999, (ACSAC '99), PROCEEDINGS, 15TH ANNUAL PHOENIX, AZ, USA 6-10 DEC. 1999, LOS ALAMITOS, CA, USA, IEEE COMPUT. SOC, US, 6.12.1999 (1999-12-06), Seiten 209-218, XP010368586 ISBN: 0-7695-0346-2, beschreibt Techniken zur präzisen Übersetzung von globalen Richtlinienregeln zu tatsächlichen Pro-Einrichtung-Konfigurationen.

Kurze Darstellung der Erfindung

[0016] Ein Verfahren, eine Vorrichtung und ein computerlesbares Medium gemäß der Erfindung werden in den unabhängigen Ansprüchen definiert. Bevorzugte Formen werden in den abhängigen Ansprüchen definiert.

[0017] Im allgemeinen werden ein Verfahren und eine Vorrichtung zum Analysieren der Funktionsweise einer oder mehrerer Firewalls oder anderer Netzwerk-Gateways wie zum Beispiel von Routern, die eine Paketfilterungsfunktion in einer Netzwerkumgebung durchführen, offengelegt. Die Sicherheitsrichtlinie für eine bestimmte Netzwerkumgebung wird typischerweise durch Definieren einer Paketfilterungskonfigurationsdatei (einer Regelbasis) für jede Firewall implementiert. Die Paketfilterungskonfigurationsdatei instruiert eine gegebene Firewall, ob sie bestimmte ankommende und abgehende Pakete durchlassen oder auskoppeln soll. Wenn eine Benutzeranfrage gegeben ist, simuliert das offengelegte Firewall-Analysewerkzeug das verhalten der verschiedenen Firewalls, wobei die Topologie der Netzwerkumgebung berücksichtigt wird, und bestimmt, welche Teile der Dienste oder Maschinen, die in der ursprünglichen Anfrage spezifiziert wurden, in der Lage sein würden, von der Quelle zum Ziel zu reichen.

[0018] Das offengelegte Firewall-Analysewerkzeug sammelt und liest die relevanten Paketfilterungskonfigurationsdateien und konstruiert eine interne Repräsentation der implizierten Sicherheitsrichtlinien. Zusätzlich verwendet das Firewall-Analysewerkzeug eine Graph-Datenstruktur zur Repräsentation der Netzwerktopologie. Ein Gateway-Zonen-Graph besteht aus einer Anzahl von durch Kanten verbundenen Knoten. Jeder Knoten entspricht entweder einem Gateway (Firewall oder Router) oder einer durch ein Gateway erzeugten Zone. Im allgemeinen nimmt die vorliegende Erfindung an, daß ein gegebenes Paket auf einem beliebigen physischen Weg wandern kann, auch wenn er gemäß dem Routenschema nicht erlaubt ist. Der Gateway-Zonen-Graph enthält einen Knoten für jede Einrichtung, die eine Paketfilterungs-Regelbasis enthält, und für jede durch solche Einrichtungen definierte Zone. Das Firewall-Analysewerkzeug muß nicht jeden Router und Switch in dem Netzwerk kennen und ist gegenüber dem verwendeten Routing-Schema indifferent.

[0019] Durch den Gateway-Zonen-Graph kann das Firewall-Analysewerkzeug bestimmen, wo gegebene Pakete in dem Netzwerk wandern und welche Gateways auf diesen Wegen angetroffen werden. Auf diese Weise kann das Firewall-Analysewerkzeug ein Anfrageobjekt im Vergleich zu jedem Regelbasisobjekt für jeden Gateway-Knoten in dem Gateway-Zonen-Graph, der auf jedem Weg zwischen Quelle und Ziel angetroffen wird, auswerten.

[0020] Gemäß einem weiteren Aspekt der Erfindung liefert das Firewall-Analysewerkzeug eine graphische Benutzeroberfläche zum Empfangen und Auswerten von einfachen Anfragen, wie zum Beispiel ob einer oder mehrere gegebene Dienste zwischen einer oder mehreren gegebenen Maschinen erlaubt sind. Durch die vorliegende Erfindung kann ein Benutzer Anfragen aggregieren, wenn ein gegebener Dienst möglicherweise eine Menge von Diensten ist (bis hin zu einer Wildcard „alle möglichen Dienste"), und gegebene Maschinen können beliebige Mengen von IP-Adressen sein (bis zu einer Wildcard „alle möglichen Adressen"). Gemäß einem weiteren Merkmal der vorliegenden Erfindung kann das Firewall-Analysewerkzeug eine Spoofing-Attacke durch Verändern der Quellen-IP-Adresse simulieren. Durch das Firewall-Analysewerkzeug kann der Benutzer spezifizieren, wo die Pakete in das Netzwerk eingespeist werden sollen, wobei es sich möglicherweise nicht um den wahren Ort der Quellen-Hostgruppe handelt. Das Firewall-Analysewerkzeug kann außerdem Firewall-Regeln berücksichtigen, die Netzwerkadressenübersetzung (NAT) durchführen.

[0021] Ein vollständigeres Verständnis der vorliegenden Erfindung sowie weitere Merkmale und Vorteile der vorliegenden Erfindung erhält man durch Bezugnahme auf die folgende ausführliche Beschreibung und die Zeichnungen.

Kurze Beschreibung der Zeichnungen

[0022] Fig. 1 zeigt eine repräsentative Netzwerkumgebung gemäß der vorliegenden Erfindung;

[0023] Fig. 2 zeigt die Komponenten des Firewall-Analysewerkzeugs von Fig. 1;

[0024] Fig. 3 zeigt einen Gateway-Zonen-Graph der Netzwerkumgebung von Fig. 1;

[0025] Fig. 4 ist ein Flußdiagramm eines durch das Firewall-Analysewerkzeug von Fig. 2 durchgeführten beispielhaften Anfrage-Engine-Algorithmus; und

[0026] Fig. 5 bis Fig. 7 sind Ansichten einer von dem Firewall-Analysewerkzeug von Fig. 2 zum Empfangen einer Benutzeranfrage und zum Angeben von Ergebnissen verwendeten beispielhaften graphischen Benutzeroberfläche.

Ausführliche Beschreibung

[0027] Fig. 1 zeigt eine repräsentative Netzwerkumgebung **100** gemäß der vorliegenden Erfindung. Wie in Fig. 1 gezeigt, enthält das Netzwerk **100** zwei Firewalls **120**, **150**. Die externe Firewall **120** schützt die Verbindung der Firma mit einem externen Netzwerk, wie zum Beispiel dem Internet **110**. Hinter der externen Firewall **120** befindet sich die Serverzone **130**, die häufig als die „demilitarisierte Zone" (DMZ) bezeichnet wird, und die die extern sichtbaren Server der Firma enthält. Bei der beispielhaften Ausführungsform umfassen die sichtbaren Server in der Serverzone **130** einen mehrfachen Server **138**, der folgendes enthält: Dienste für e-Mail (smtp), http-Dateitransfers (web) (hypertext transfer protocol) und ftp-Dateitransfers (file transfer protocol) und einen Domänennamenserver (dns) **134**.

[0028] Hinter der Serverzone befindet sich eine interne Firewall **150**, die das proprietäre oder interne Netzwerk der Firma, wie zum Beispiel ein Intranet, schützt. Die interne Firewall **150** besitzt drei Schnittstellen. Eine erste Schnittstelle ist an der Serverzone **130**, eine zweite Schnittstelle verbindet die interne Firewall **150** mit der Firmennetzwerkzone **160** und eine dritte Schnittstelle verbindet die interne Firewall **150** mit der Firewall-Administrationszone **140**. Die Sicherung des Firewall-Administrationshosts ist für die Integrität des Netzwerks kritisch und sollte von den anderen Firmenhosts getrennt sein. Innerhalb der Firmennetzwerkzone befindet sich im allgemeinen ein distinguierter Host, der als (nicht gezeigter) Steuerhost bezeichnet wird und die Administration für die Server in der Serverzone **130** gewährleistet. Bei der beispielhaften Ausführungsform ist mit jeder Firewall **120**, **150** eine Paketfilterungskonfigurationsdatei **125**, **155**, die nachfolgend besprochen wird, assoziiert.

[0029] Die Paketfilterungskonfigurationsdateien **125**, **155** sind im allgemeinen firewallspezifische Regelbasen. In der Mehrfach-Firewall-Umgebung von Fig. 1 unterteilen die verschiedenen Firewalls die Intranets der Firma in mehrere Zonen, wie zum Beispiel die Serverzone (demilitarisierte Zone – DMZ) **130**, die Firewall-Administrationszone **140** und die Firmennetzwerkzone **160**. In diesem Fall wird die Sicherheitsrichtlinie typischerweise durch mehrere Regelbasen realisiert, die sich auf den verschiedenen Gateways befinden. Das Zusammenspiel zwischen diesen Regelbasen bestimmt also, welche Sitzungen durchgelassen werden.

[0030] Die beispielhafte Firmennetzwerkumgebung von Fig. 1 verwendet eine Sicherheitsrichtlinie, die den vertrauenswürdigen internen Firmenbenutzern uneingeschränkten Zugang gibt, während externe Benutzer nur auf Inhalt zugreifen dürfen, der explizit öffentlich zugänglich gemacht wird. Genauer gesagt erlaubt es die Sicherheitsrichtlinie internen Firmenhosts, auf alle Betriebsmittel auf dem Internet zuzugreifen. Externe Hosts können jedoch nur auf die Server in der Serverzone **130** zugreifen. Insbesondere werden smtp-Dienste für Firmenbenutzer nur über den Mail-Server **138** zugelassen und DNS-Dienste werden für das Internet nur durch den DNS-Server **134** bereitgestellt. Außerdem können die Server **134**, **138** nur durch die (nicht gezeigte) Webadministratorhoststeuerung aktualisiert werden, die die Administration für die Server in der Serverzone **130** gewährleistet. Andere Firmenbenutzer haben dieselben Privilegien wie Internet-Hosts in bezug auf die DMZ-Server **134**, **138**. Schließlich sind die Firewall-Schnittstellen **120**, **150** nur von dem Firewall-Administratorhost **140** aus zugänglich.

[0031] Wie in Fig. 1 gezeigt, liefert die vorliegende Erfindung ein Firewall-Analysewerkzeug **200**, das alle relevanten Paketfilterungskonfigurationsdateien **125**, **155** sammelt und liest und eine interne Repräsentation der implizierten Sicherheitsrichtlinie und der Netzwerktopologie konstruiert, wie später ausführlicher besprochen werden wird. Das Firewall-Analysewerkzeug **200** liefert eine graphische Benutzeroberfläche zur Auswertung

einfacherer Anfragen, wie zum Beispiel ob ein oder mehrere gegebene Dienste zwischen einer oder mehreren gegebenen Maschinen zugelassen sind. Ein Benutzer kann Anfragen aggregieren, wobei ein gegebener Dienst eine Menge von Diensten sein kann (bis zu einer Wildcard „alle möglichen Dienste") und gegebene Maschinen beliebige Mengen von IP-Adressen sein können (bis zu einer Wildcard „alle möglichen Adressen"). Wenn eine Anfrage gegeben ist, simuliert das Firewall-Analysewerkzeug **200** das Verhalten der verschiedenen Firewalls **120**, **150**, wobei die Topologie der Netzwerkumgebung **100** berücksichtigt wird, und berechnet, welche Teile der Dienste oder Maschinen, die in der ursprünglichen Anfrage spezifiziert waren, von Quelle zum Ziel reichen könnten. Das Firewall-Analysewerkzeug **200** bestimmt also, ob nur eine Teilmenge der Dienste zugelassen ist oder ob solche zugelassenen Dienste nur zwischen Teilmengen der spezifizierten Quellen- und Ziel-Hostgruppen zugelassen sind.

[0032] Gemäß einem weiteren Merkmal der vorliegenden Erfindung kann das Firewall-Analysewerkzeug **200** Spoofing-Attacken simulieren, die die Quellen-IP-Adresse verändern. Das Firewall-Analysewerkzeug **200** erlaubt dem Benutzer, zu spezifizieren, wo die Pakete in das Netzwerk eingespeist werden sollen, wobei es sich möglicherweise nicht um den wahren Ort der Quellen-Hostgruppe handelt. Das Firewall-Analysewerkzeug **200** kann außerdem Firewall-Regeln berücksichtigen, die Netzwerkadressenübersetzung (NAT) durchführen.

Firewall-Terminologie und -Modellierungskonzepte

[0033] Ein Firewall-Konfigurationswerkzeug ermöglicht es dem Sicherheitsadministrator typischerweise, verschiedene Hostgruppen (Ansammlungen von IP-Adressen) und Dienstgruppen (Gruppen von Protokollen und entsprechenden Portnummern an den Hosts, die die Endpunkte bilden), zu definieren. Eine einzelne Regel enthält typischerweise eine Quelle, ein Ziel, eine Dienstgruppe und eine entsprechende Aktion. Quelle und Ziel sind Hostgruppen und die Aktion ist typischerweise entweder das „Durchlassen" oder „Auskoppeln" der Pakete der entsprechenden Sitzung. Zusätzlich kann die Aktion das Schreiben eines Protokollierungsdatensatzes oder das Durchführen einer Netzwerkadressenübersetzung (NAT) spezifizieren.

[0034] Wie bereits erwähnt, ist die Regelbasis häufig reihenfolgeabhängig. Im allgemeinen prüft die Firewall, ob die erste Regel in der Regelbasis für eine neue Sitzung gilt. Wenn dem so ist, werden die Pakete gemäß der Aktion der ersten Regel entweder ausgekoppelt oder durchgelassen. Andernfalls prüft die Firewall, ob die zweite Regel gilt und so weiter.

[0035] Im vorliegenden Kontext sind Gateways die Paketfilterungsmaschinen und können entweder Firewalls oder Router sein. Normalerweise hat ein Gateway mehrere Heimaten, da ein Gateway mindestens zwei Internetverbindungen aufweist. Typischerweise besitzt ein Gateway mehrere Netzwerkverbindungen. Jede Verbindung geht durch eine Schnittstelle, die ihre eigene einmalige IP-Adresse besitzt. Es wird angenommen, daß mit jeder Schnittstelle eine Paketfilterungskonfigurationsdatei **125**, **155** assoziiert ist. Die Gateways teilen den IP-Adressenraum in elementfremde Zonen auf, wie in Fig. 1 gezeigt. Genauer gesagt ist eine Zone z eine maximale Menge von IP-Adressen dergestalt, daß zwischen beliebigen zwei Adressen in z gesendete Pakete kein Filterungsgateway durchlaufen. Die meisten Zonen entsprechen einem Subnet(s) einer Firma, wobei gewöhnlich eine große „Internet"-Zone **110** dem Teil des IP-Adressenraums entspricht, der nicht von der Firma verwendet wird.

[0036] Ein Dienst ist die Kombination einer Protokollbasis wie zum Beispiel tcp oder udp und der Portnummern sowohl auf der Quellen- als auch auf der Zielseite. Zum Beispiel ist der Dienst telnet als tcp mit Zielport 23 und einem beliebigen Quellenport definiert. Eine Dienstgruppe ist einfach eine Menge von Diensten.

FIREWALL-ANALYSEWERKZEUG

[0037] Wie in Fig. 2 gezeigt, erfordert das Firewall-Analysewerkzeug **200** ein instantiiertes Modell der Netzwerktopologie, das in einer benutzergeschriebenen Topologiedefinitionsdatei **210** beschrieben wird, die später ausführlicher in dem Abschnitt mit dem Titel „Topologiedatei" besprochen wird. Wie in Fig. 2 gezeigt, wird die Topologiedefinitionsdatei **210** unter Verwendung einer Teilmenge der Modelldefinitionssprache (MDL) geschrieben, die nachfolgend in einem Abschnitt mit dem Titel „Modelldefinitionssprache (MDL)" besprochen wird.

[0038] Wie in Fig. 2 gezeigt, verwendet die Anfrage-Engine **240** des Firewall-Analysewerkzeugs **200** eine Kombination einer nachfolgend in Verbindung mit Fig. 3 besprochenen Graph-Datenstruktur und eines in einem Anfragealgorithmus **400**, der nachfolgend in Verbindung mit Fig. 4 besprochen wird, enthaltenen Regelbasissimulators. Wie in Fig. 2 gezeigt, nimmt die Anfrage-Engine **240** eine Anfrage von einem Benutzer **220**

als Eingabe, die aus einer Quelle und einem Ziel (die beide Hostgruppen sind) und einer Dienstgruppe besteht.

[0039] Gemäß einem Merkmal der vorliegenden Erfindung muß das Firewall-Analysewerkzeug **200** nicht jeden Router und Switch in dem Netzwerk kennen und ist gegenüber dem verwendeten Routing-Schema indifferent. Im allgemeinen nimmt die vorliegende Erfindung an, daß sich ein gegebenes Paket auf einem beliebigen physischen Weg ausbreiten kann, auch wenn er gemäß dem Routenschema nicht erlaubt ist. Die vorliegende Erfindung betrachtet die Einrichtungen, auf denen Paketfilterungsregelbasen installiert sind, und die von diesen Einrichtungen definierten Zonen. Auf dieser Granularitätsebene ist die Topologie relativ stabil. Die Topologiedatei **210** muß also nur dann modifiziert werden, wenn die Firewalls **120**, **150** zu dem Netzwerk **100** hinzugefügt oder darin ausgewechselt werden.

[0040] Als Teil der Topologiedefinitionsdatei **210** spezifiziert der Benutzer **220** die Namen der Paketfilterungskonfigurationsdateien **125**, **155** (die im folgenden kollektiv als Firewall-Konfigurationsdateien **230** bezeichnet werden), die die Regelbasen für alle Gateway-Schnittstellen **120**, **150** in der Netzwerkumgebung **100** enthalten. Nach dem Lesen der Topologiedefinitionsdatei **210** analysiert das Firewall-Analysewerkzeug **200** jede dieser Konfigurationsdateien **230** der Reihe nach (unter Verwendung eines separaten „Frontend"-Moduls für jede unterstützte Firewall-Variante) und füllt seine internen Regelbasisdatenstrukturen für jede Einrichtung auf. Es wird wieder angemerkt, daß diese Firewall-Konfigurationsdateien **230** vertreiberspezifisch sind und von jeglichen Werkzeugen erzeugt werden, die zur Konfiguration der fraglichen Einrichtungen verwendet werden.

[0041] Der Benutzer bildet eine Anfrage durch Wählen jedes Elements des Anfrage-Tripels (Dienst, Quelle, Ziel), wie zum Beispiel aus einem Drop-Down-Menü, das Wahlmöglichkeiten aller Hostgruppen oder Dienstgruppen, die in den Konfigurationsdateien definiert wurden, anbietet.

Topologiemodellierung

[0042] Die Netzwerktopologie wird durch Aufteilen des Netzwerks **100** in Zonen **130**, **140**, **160** modelliert, die durch Gateways verbunden sind. Ein Gateway **120**, **150** besitzt eine Schnittstelle für jede angrenzende Zone **130**, **140**, **160**. Jede Schnittstelle besitzt ihre eigene IP-Adresse (und wird für bestimmte Zwecke als Host betrachtet) oder wird als unsichtbar deklariert (zum Beispiel unter Verwendung eines INVIS-Schlüsselworts), wenn die Firewall **120**, **150** als Brücke arbeitet. Pakete, die eine Zone **130**, **140**, **160** verlassen und in diese eintreten, können durch das Gateway **120**, **150** an der entsprechenden Schnittstelle gefiltert werden. Pakete, die innerhalb derselben Zone **130**, **140**, **160** gesendet und empfangen werden, können nicht durch ein Gateway **120**, **150** gefiltert werden, und zwar einfach deshalb, weil die Pakete kein Gateway **120**, **150** durchlaufen. Vom Standpunkt des Firewall-Analysewerkzeugs **200** aus gesehen existiert also ein Weg zwischen zwei beliebigen Hosts in derselben Zone, wobei beliebige und alle Filterung durch die Schnittstelle durchgeführt wird. Zonen bestehen aus Hostgruppen. Hostgruppen werden typischerweise weiter in eine Hierarchie kleinerer Hostgruppen oder einzelner Hosts unterteilt.

Topologiedatei

[0043] In diesem Abschnitt wird eine vollständige Auflistung der Topologiedatei gegeben, die in MDL geschrieben ist und zur Beschreibung der Netzwerkumgebung **100** dient. Die Topologiedatei liefert einen Eingabemechanismus zur Konstruktion des Gateway-Zonen-Graph **300**, der später ausführlicher besprochen wird. Als erstes werden die Hostgruppen definiert:

```
HOST_GROUPS {

    # die Zonen

    Z_dmz   = [111.222.1.0/24]

    Z_corp  = [111.222.2.0/24]

    Z_admin = [111.222.3.0/24]

    Z_internet = [0.0.0.0    - 111.221.255.255,

                  111.222.100.0 - 255.255.255.255]

    # die (sichtbaren) Gateway-Schnittstellen

    I_dmz_in    = [111.222.1.1]

    I_admin     = [111.222.3.1]

}
```

[0044] Es wird angemerkt, daß ein IP-Adressenbereich auf verschiedene Weisen spezifiziert werden kann. Die Schrägstrichnotation definiert einen Bereich durch Angabe, wie viele der höchstwertigen Bit fest sind.

[0045] Als nächstes werden die Schnittstellen definiert.

```
SCHNITTSTELLEN {

        I_internet_dmz = { INVIS, NO_GEN }

    I_dmz_in     = { file="~/fw/analyzer/data/dmz_in" }

    I_dmz_corp   = { INVIS, NO_GEN }

    I_corp_in    = { INVIS,

                   file="~/fw/analyzer/data/corp_in" }

    I_admin      = { file="~/fw/analyzer/data/admin" }

}
```

[0046] Schnittstellen mit dem Attribut NO GEN führen keine Filterung durch. Schnittstellen mit dem Attribut INVIS besitzen keine IP-Adresse, da sie zu einer Firewall gehören, die als Brücke arbeitet.

[0047] Als letztes werden die Gateways und Zonen definiert.

```
GATEWAYS {

    dmz_gw  = {I_internet_dmz, I_dmz_in}      : LMF

    corp_gw = {I_dmz_corp, I_corp_in, I_admin} : LMF

}

ZONES {

    Z_internet = { I_internet_dmz }

    Z_dmz      = { I_dmz_in, I_dmz_corp }

    Z_corp     = { I_corp_in }

    Z_admin    = { I_admin }

}
```

Benennung

[0048] In dem von der vorliegenden Erfindung benutzten Modell besitzen alle Objekte (Hosts, Hostgruppen, Dienstgruppen) im allgemeinen Namen. Dadurch wird ein hoher Grad an Abstraktion bei der Interaktion mit dem Benutzer gewährleistet. Sinnvolle Namen drücken mehr aus als rohe IP-Adressen und Portnummern. Soweit wie möglich erhält das Firewall-Analysewerkzeug **200** diese Namen von den vertreiberspezifischen Konfigurationsdateien. Da jedoch angenommen wird, daß jede Einrichtung und Schnittstelle unabhängig konfiguriert wurde, können Namenkonflikte bestehen. Zum Beispiel kann der Administrator den Namen http als tcp auf Port 80 auf einem Gateway bedeutend definiert haben, während auf einem anderen Gateway der Administrator möglicherweise denselben Namen für tcp auf den Ports 80, 8000 und 8080 verwendet. Um dieses Niveau der Benennung zu unterstützen, führt das Firewall-Analysewerkzeug **200** einen separaten Symboltabellenkontext pro Schnittstelle (d.h. pro Regelbasis). Wenn derselbe Name in verschiedenen Kontexten mit verschiedenen Bedeutungen erscheint, kann das Firewall-Analysewerkzeug **200** wahlweise alle Varianten in seinen Drop-Down-Menüs, mit vorangestelltem Schnittstellennamen, zeigen. Wenn dagegen alle Varianten identisch sind, erscheint der Name nur einmal ohne Präfix.

Regelbasen

[0049] Die innerhalb der Firmennetzwerkumgebung **100** eingerichtete implizite Sicherheitsrichtlinie wird von den vertreiberspezifischen Paketfilterungskonfigurationsdateien **125**, **155** abgeleitet. Das Firewall-Analysewekzeug **200** transformiert jede dieser Paketfilterungskonfigurationsdateien **125**, **155**, die mit einer Schnittstelle assoziiert sind, zu einer Tabelle logischer Regeln, die die folgende Datensatzstruktur enthält (zur leichteren Erläuterung vereinfacht):

```
struct rule {
  struct hostgrp      *source;
  struct hostgrp      *dest;
  struct servicegrp   *service_grp;
  direction_ty        direction;
  action_ty           action;
};
```

[0050] Die tatsächliche Semantik ist von Vertreiber zu Vertreiber unterschiedlich. Bei der beispielhaften Ausführungsform wird die folgende beispielhafte Semantik verwendet. Wenn Pakete gefiltert werden, werden die Regeln in der Liste in ihrer Reihenfolge untersucht, bis eine Übereinstimmung auftritt. Die Felder Quelle, Ziel und service_grp werden mit den entsprechenden Feldern in dem Paket verglichen. „Direction" spezifiziert, ob die Regel für in das Gateway **120**, **150** eintretende Pakete (IN) oder dieses verlassende Pakete (OUT) gilt, auf der diese Schnittstelle verankert ist (d.h. die Regeln sind Gateway-zentrisch). Die Wildcard-Richtung (BOTH) gibt an, daß die Regel für beide Richtungen gilt. Wenn eine Übereinstimmung auftritt, wird die entsprechende

Aktion (DROP oder PASS) durchgeführt. Die interne Regelbasistabelle unterstützt außerdem Regeln, die NAT durchführen, und die Regelstruktur besitzt daher einige zusätzliche Felder.

Anfragen

[0051] Ein zentrales Objekt in dem Firewall-Analysewerkzeug **200** ist eine Anfrage. Eine Anfrage ist ein Tripel, das aus einer Quellen-Hostgruppe, einer Ziel-Hostgruppe und einer Dienstgruppe besteht. Die Semantik einer solchen Anfrage lautet: „Welche IP-Adressen in der Quellen-Hostgruppe können Dienste aus der Dienstgruppe zu welchen IP-Adressen in der Ziel-Hostgruppe senden?". Die Anfrage wird durch die folgende Datenstruktur beschrieben:

```
struct query {

  struct hostgrp *src;

  struct hostgrp *dst;

  struct servicegrp *service;

};
```

[0052] Es wird wieder angemerkt, daß Hostgruppen und Dienstgruppen Wildcards sein können, d.h. ein beliebiges Element des Anfragetripels kann die Wildcard „*" sein, das „beliebig" bedeutet. Die Frage „welche Maschinen können die Webserver der Firma benutzen?" kann also durch das Anfragetripel mit der Form (*, web_servers, http_services) ausgedrückt werden, wobei angenommen wird, daß die Hostgruppe web_servers und die Dienstgruppe http_services definiert sind.

[0053] Typischerweise können nicht alle durch eine Anfrage beschriebenen Pakete das Ziel erreichen. Durch die Operationen der verschiedenen Regelbasen **125**, **155** können bestimmte Pakete ausgekoppelt werden. Das Firewall-Analysewerkzeug **200** beantwortet deshalb eine solche Anfrage mit einer verfeinerten Liste von „Subanfragen", d.h. einer Liste von Anfragetripeln, wobei jedes Element eine Teilmenge des entsprechenden Elements in dem Anfragetripel ist.

[0054] Die Semantik der Antwort besteht darin, daß für jedes Teilmengentripel die entsprechende Quellen-Hostgruppe tatsächlich den Dienst zu der Ziel-Hostgruppe senden kann.

Gateway-Zonen-Graph

[0055] Das Firewall-Analysewerkzeug **200** verwendet eine Graph-Datenstruktur zur Repräsentation der Netzwerktopologie. Fig. 3 zeigt einen Gateway-Zonen-Graph **300** für die Netzwerkumgebung von Fig. 1. Der Gateway-Zonen-Graph **300** besteht aus einer Anzahl von durch Kanten **321**-**325** verbundenen Knoten **311**-**316**. Eine ausführliche Besprechung der Erzeugung von Gateway-Zonen-Graphen **300** gemäß graphtheoretischen Prinzipien findet man zum Beispiel in T. H. Cormen et al., „Introduction to Algorithms", 463-630 (MIT Press, 1989). Der in Fig. 3 gezeigte beispielhafte Gateway-Zonen-Graph **300** verwendet die folgende Standardnotation. Das Dreiecksymbol „Δ" gibt ein Gateway **120**, **150** in dem Netzwerk **100** an, und das Rechtecksymbol „□" eine Zone **110**, **130**, **140**, **160** in dem Netzwerk **100**. Zwischen zwei Knoten **311**-**316** besteht eine Kante **321**-**325**, wenn eine physische Verbindung zwischen ihnen besteht. Im allgemeinen ermöglicht es der Gateway-Zonen-Graph **300** dem Firewall-Analysewerkzeug **200** zu bestimmen, wohin sich gegebene Pakete in dem Netzwerk **100** ausbreiten werden und welche Gateways **120**, **150** auf diesen Wegen angetroffen werden.

[0056] Zu diesem Zweck enthält das interne Modell den folgenden Hilfs-Graph. Im vorliegenden Kontext ist der Gateway-Zonen-Graph **300** als ein zweigeteilter Graph $H=((G \cup Z), I)$ definiert, dessen Ecken aus der Menge von Gateways G und der Menge Zonen Z bestehen. Die Menge der Schnittstellen I bildet die Kanten: H enthält eine Kante $i=(g,z)$, die ein Gateway $g \in G$ mit einer Zone $z \in Z$ verbindet, genau dann, wenn g eine Schnittstelle i aufweist, deren angrenzende Zone z ist.

[0057] Die Ecken des Gateway-Zonen-Graph **300** werden mit der folgenden Struktur implementiert.

```
struct node {

        union {

          struct zone *z;

          struct gateway *gw;

        } zg;

        struct hostgrp *hg;

        node_ty type;

        struct query *q;

      };
```

[0058] Für jedes Gateway und jede Zone gibt es einen Knoten. Das Typenfeld gibt an, ob ein Gateway oder eine Zone durch einen gegebenen Knoten repräsentiert wird. Das hg-Feld führt den in dem Knoten enthaltenen IP-Bereich.

[0059] Für Zonenknoten ist hg die Hostgruppe der Zone minus die IP-Adressen der angrenzenden Schnittstellen dieser Zone. Für Gateway-Knoten ist hg die Menge der IP-Adressen der an das Gateway angeschlossenen Schnittstellen. Das q-Feld in der Knotenstruktur wird für die Verarbeitung der Anfrage auf die nachfolgend besprochene Weise verwendet.

Anfrage-Engine-Algorithmus

[0060] Die Anfrage-Engine des Firewall-Analysewerkzeugs **200** verwendet eine Kombination der oben in Verbindung mit Fig. 3 besprochenen Graph-Datenstruktur und eines in dem Anfragealgorithmus **400**, der nachfolgend in Verbindung mit Fig. 4 besprochen wird, enthaltenen Regelbasissimulators. Wie in Fig. 2 gezeigt, nimmt die Anfrage-Engine **240** als Eingabe eine Anfrage von einem Benutzer **220** an, die aus Quelle und Ziel (beides Hostgruppen) und einer Dienstgruppe besteht. Dann simuliert sie das Verhalten aller durch die Anfrage beschriebenen Pakete, während sie das Netzwerk durchqueren.

[0061] Fig. 4 ist ein Flußdiagramm eines beispielhaften durch das Firewall-Analysewerkzeug von Fig. 2 durchgeführten Anfrage-Engine-Algorithmus **400**. Wie in Fig. 4 gezeigt, wird im Schritt **410** eine Benutzeranfrage empfangen und die Anfrage wird anfangs an den Knoten in dem Gateway-Zonen-Graph angebunden, der die Quellen-Hostgruppe enthält (Schritt **420**). Wenn die Quellen-Hostgruppe nicht in einer einzigen Zone enthalten ist (wenn zum Beispiel die Wildcard * benutzt wird), wird die Quellen-Hostgruppe im Schritt **420** in elementfremde Hostgruppen aufgeteilt, von denen jede in einer Zone enthalten ist. Dann wird in Schritt **430** für jede Quellen-Hostgruppe eine getrennte Graph-Suche durchgeführt. Die Graph-Suche wertet im allgemeinen das Anfrageobjekt im Vergleich zu jedem oben besprochenen Regelbasisobjekt für jeden Gateway-Knoten in dem Gateway-Zonen-Graph **300**, der bei der Graph-Suche angeroffen wird, aus.

[0062] Dann versucht der Algorithmus, die Anfrage über alle Kanten zu propagieren, die den aktuellen Knoten verbinden (Schritt **440**). Der Algorithmus wird auf dieselbe Weise fortgesetzt, wobei die Anfrage weiter propagiert wird, bis sie den gesamten Graph durchsucht. Alle die Anfrage erfüllenden Tripel werden im Schritt **450** identifiziert, bevor die Programmsteuerung endet (Schritt **460**).

[0063] Der grundlegende Schritt des Algorithmus **400** ist die Ausbreitung einer Anfrage über eine Kante in dem Gateway-Zonen-Graph **300**, wodurch eine Firewall-Schnittstelle repräsentiert wird. Dadurch wird die Auswirkung der an die Schnittstelle angeschlossenen Regelbasis auf die durch die Anfrage beschriebenen Pakete modelliert. Typischerweise können nur Teile der Anfragen eine beliebige gegebene Kante überqueren, da ein Teil der Pakete durch die Schnittstelle ausgekoppelt werden würde. Nach dem Überqueren einer Kante muß deshalb möglicherweise die Anfrage in eine Menge verfeinerter Anfragen zerlegt werden, die nur die Pakete repräsentieren, die durchgelassen worden wären. Zum Beispiel kann die ursprüngliche Anfrage (corporate_net, internet, *) gewesen sein, aber die Regelbasis erlaubt nur abgehende http- und smtp-Dienste, so daß die Menge von Anfragen, die die andere Seite der Kante erreicht, nun (corporate net, internet, tcp), (corporate_net, internet, smtp) ist.

[0064] Es wird angemerkt, daß bestimmte Knoten möglicherweise mehr als einmal besucht werden, während andere Knoten überhaupt nicht besucht werden, da der Algorithmus **400** über alle möglichen Wege rückverfolgt, die die Anfrage durch das Netzwerk **100** nehmen kann und fortgesetzt wird, solange ein bestimmter Teil der Anfrage unausgekoppelt bleibt. Wenn die Anfrage einen Knoten v (Zone oder Gateway) über verschiedene Wege erreichen kann, ist die neue Anfrage, die an v angebunden ist, die Vereinigungsmenge der Anfrageergebnisse, die v auf jedem der möglichen Wege erreichen. Bei einer Variante kann die Suche optimiert werden, um eine Überquerung nicht durchzuführen, wenn es klar ist, daß keine neuen Pakete durchgelassen werden, die nicht schon bei anderen Wegen gewesen sind.

[0065] Die letzte Phase der Verarbeitung einer Anfrage ist das Sammeln der Ergebnisse (Schritt **450**). Dabei wird einfach der Knoten bzw. werden einfach die Knoten betrachtet, der/die die Ziel-Hostgruppe enthalten und es werden die Anfragen herausgesucht, die ihr korrektes Ziel erreicht haben.

[0066] Im ungünstigsten Fall ist die Komplexität des Algorithmus **400** exponentiell in der Größe des Gateway-Zonen-Graphen **300**. Dieser ungünstigste Fall kann jedoch nur in sehr dichten Graphen auftreten. Typische Gateway-Zonen-Graphen **300** sind sehr spärlich, da die Firewalls **120**, **150** normalerweise an strategischen Drosselpunkten in dem Netzwerk plaziert sind und die am häufigsten anzutreffende Gateway-Zonen-Graph-Topologie ein Baum ist. Auf einer Baumtopologie ist der Algorithmus im wesentlichen eine Depth-First-Suche, d.h. er ist linear in der Größe des Graphen. Da nur Zonen, die durch Firewalls getrennt sind, modelliert werden, sind die Gateway-Zonen-Graphen außerdem tendenziell relativ klein.

Spoofing

[0067] Eine kleine Erweiterung des grundlegenden Algorithmus **400** ermöglicht ein Prüfen auf Spoofing-Attacken. Zusätzlich zu den Parametern Quelle, Ziel und Dienst, die eine Anfrage definieren, wird ein optionaler Filterparameter hinzugefügt, der die wahre Quelle spezifiziert, von der die Pakete stammen. Wenn dieser vierte Parameter definiert wird, versteht sich, daß die ursprüngliche Quellen-Hostgruppe die falsche Quellenadresse in dem Paket ist. Die Anfrage kann dann auf die oben beschriebene Weise verarbeitet werden, mit der Ausnahme, daß, statt von den Knoten zu starten, die eine falsche Quellen-Hostgruppe enthalten, der Algorithmus an den Knoten startet, die die wahre Quelle enthalten.

Beispiel

[0068] Man betrachte die Anfrage: „Welche Dienste sind zwischen der Firmenzone und der DMZ erlaubt?" Die in Fig. 5 gezeigten Ergebnisse geben wieder, daß die Firewall-Konfiguration in dieser Hinsicht korrekt ausgeführt wurde. Die Dienste sind allen Hosts in der Firmenzone verfügbar, aber nur die Steuerung kann tcp-Verbindungen zu den Servern öffnen. Außerdem ist zu bemerken, daß nur die Namen der Host- und Dienstgruppen angezeigt sind, aber durch Erweitern eines Eintrags (wahlweise über Mausklicken) weitere Einzelheiten erhältlich sind (wie zum Beispiel tatsächliche IP-Adresse und Portnummern).

[0069] In einem weiteren Beispiel betrachte man die Anfrage: „Wieviel Zugang hat das Internet **110** zu dem internen Netzwerk **130**, **140**, **150**?" die Ergebnisse sind in Fig. 6 gezeigt. Die ersten fünf Linien der Ergebnisse zeigen, daß jeder beliebige Host im Internet **110** einen bestimmten eingeschränkten Zugriff auf die Server in der DMZ **130** besitzt. Vom Standpunkt des Firewall-Analysewerkzeugs **200** aus gesehen kann ferner jeder Host in der Internetzone **110** mit jedem anderen Host in dieser Zone **110** sprechen, wodurch die dritte Zeile erklärt wird. Die letzte Zeile zeigt jedoch eine Schwäche in der Implementierung der Sicherheitsrichtlinie an. Diese Zeile zeigt, daß jeder beliebige Host im Internet **110** potentiell einen beliebigen Dienst in der inneren Schnittstelle des externen Gateway öffnen kann. Eine Untersuchung der Topologiedatei enthüllt das Problem: Die äußere Schnittstelle I_internet_dmz führt überhaupt keine Filterung durch und nachdem Pakete durch sie in das Gateway eingetreten sind, können sie ohne jegliche Filterung mit den anderen Schnittstellen sprechen. Ein vorsichtigerer Ansatz, der diese Anfälligkeit auflöst, besteht darin, die Regelbasis nicht an die interne, sondern an die äußere Schnittstelle anzubinden.

[0070] Ein weiteres Beispiel zeigt, wie auf Spoofing geprüft werden kann. Der heikelste Host ist wahrscheinlich der Firewall-Administrator **140**, so daß kein Internet-Host ihn erreichen können sollte, auch mit Spoofing. Dazu wird die wirkliche Quelle auf die Internet-Zone **110** gesetzt und die gegebene Quelle (die gefälschte Adresse) ist beliebig. Die Ergebnisse sind in Fig. 7 gezeigt. Das von ihnen gezeigte Leck ist tatsächlich ein Ergebnis desselben Problems, das in der vorherigen Anfrage zu sehen war. Ein Internet-Host kann eine Nachricht mit der Quellenadresse der Schnittstelle I_dmz_in erzeugen. An der äußeren Schnittstelle erfolgt keine Filterung, und sobald es eingedrungen ist, wird das Paket so betrachtet, als stammte es aus der Schnittstelle

I_dmz_in selbst und wird durchgelassen, weil es mit einer der Regeln übereinstimmt.

[0071] Bei einer weiteren Erweiterung des Anfrage-Antwort-Mechanismus der vorliegenden Erfindung werden zusätzliche Informationen über eine Anfrage angegeben. Zum Beispiel können die Anfrageergebnisse angeben, welche Regel in welcher Schnittstelle für das Durchlassen oder Auskoppeln eines bestimmten Pakets verantwortlich ist. Solche Informationen können graphisch angezeigt werden, um den Weg zu zeigen, den Pakete von Quelle zu Ziel nehmen.

[0072] Eine weitere Erweiterung ist eine Verbesserung des Firewall-Analysewerkzeugs **200**, um Topologieunabhängige Anfragen z.B. über Definieren von Zonen als „Intern" oder „Extern" zu ermöglichen. Wenn das Firewall-Analysewerkzeug **200** Anfragen von einer Datei lesen kann, wäre der Benutzer dann in der Lage, expertenerzeugte Anfragen zum Prüfen des Netzwerks auf bestimmte grundlegende Unsicherheiten zu verwenden. Die Expertenanfragen könnten wohlbekannte unsichere Ports oder Dienste abdecken und die Zugänglichkeit solcher Ports von den externen Zonen aus prüfen. Wenn neue Anfälligkeiten bekannt werden, könnten Organisationen wie zum Beispiel CERT auf ihrer Website zum Herunterladen aktualisierte Anfragedateien zur Verfügung stellen.

Die Modelldefinitionssprache (MDL)

[0073] Wie in EP-A-1 024 627 beschrieben, wird mit einer Modelldefinitionssprache (MDL) die Sicherheitsrichtlinie instantiiert und die Richtlinie auf die Topologie abgebildet. Ein Parser übersetzt ein MDL-Programm in eine Instanz des Entitätsbeziehungsmodells. Das Modell wird durch eine entsprechende Datenstruktur ausgedrückt.

MDL für die Sicherheitsrichtlinienbeschreibung

[0074] Ein Dienst wird mittels einer Anweisung der folgenden Form definiert:

```
<service-name> =

        <protocol-base> [<dest-port-no-range>, <src-port-no-range>]
```

[0075] Zum Beispiel definiert das folgende Codefragment die weitverbreiteten Dienste smtp, ssh, ping, https und einen, der alle Pakete auf tcp-Basis bezeichnet:

```
SERVICES {

        smtp    =       TCP [25]

        ssh     =       TCP [22]

        ping    =       ICMP [8,0]

        https   =       TCP [443]

        all_tcp =       TCP [*]

}
```

[0076] Dienste können durch eine Anweisung der folgenden Form zu einer Dienstgruppe ServiceGrp gruppiert werden:
<srv-grp-name> = {<service-name1>, <service-name2> ...}

[0077] Das folgende Codefragment definiert die beiden Dienstgruppen admin-to-gtwy und gtwy-to-admin:

```
SERVICE_GROUPS {

        admin-to-gtwy = {ssh, ping}

        gtwy-to-admin = {ssh, https}
```

[0078] Eine Rolle wird durch eine Anweisung der folgenden Form definiert, wobei der Pfeil auf offensichtliche Weise das Richtungsattribut definiert, role-grp-name auf Peers zeigt und svr-grp-name auf eine Dienstgruppe zeigt.

<role-name> arrow <role(-grp)-name> : <srv-grp-name>

arrow = ← || → || ↔

[0079] Das folgende Codefragment definiert die Rollen mail_server und internal_mail_server, wie oben besprochen. Die Rollen gateway-in und gateway-out modellieren die Fähigkeiten von Gateway-Schnittstellen in jeder Richtung. Dieses Beispiel wird im nächsten Codefragment fortgesetzt:

```
ROLE_DEFINITIONS {
        mail_server ↔ * : smtp
        internal_mail_server ↔ mail_server : smtp
        gateway_in ← fw_admin : admin_to_gtwy
        gateway_out → fw_admin : gtwy_to_admin
        intranet_machine → all_tcp : *
}
```

[0080] Rollen werden zu offenen (Vorgabe) Rollengruppen **325** durch die folgende Anweisung gruppiert:
<role-grp-name> = {<role-name1>, <role-name2> ...}

[0081] Rollen werden zu geschlossenen Rollengruppen durch die folgende Anweisung gruppiert:
<role-gtp-name> = «<role-name1>, <role-name2> ...»

[0082] Das folgende Codefragment definiert die Rollengruppe, Gateway, wodurch die unidirektionalen Gateway-Rollen in einer Rollengruppe gebündelt werden. Es wird angemerkt, daß die Gateway-Rollengruppe geschlossen ist, wodurch Hosts, die diese Rollengruppe annehmen, effektiv „getarnt" werden.

```
ROLE_GROUPS {
gateway     =     <<gateway_in, gateway_out>>  # eine geschlossene Gruppe
```

MDL für Topologiebeschreibung und Richtlinienabbildung

[0083] Hosts und Hostgruppen werden durch die folgenden Anweisungen definiert:
<host-name> = [<IP-Addr>] : <role-grp-name>
<host-grp-name> = [<IP-Range>] : <role-grp-name>

[0084] Das folgende Codefragment definiert die Hosts dirty (vermutlich außerhalb des Intranets) und dusty, wobei ihnen die Rollen externer bzw. interner Mail-Server zugewiesen werden:

```
HOST {
        dirty   =     [ 111.222.100.6 ]     : mail_server
        dusty   =     [ 111.222.1.3]        : internal_mail_server
}
```

[0085] Gateways werden durch die folgende Anweisung definiert:
<gateway-name> = {<host-name1>; <host-name2> ...}

[0086] Das folgende Codefragment definiert payroll_gw_interface1/2 als Hosts und spezifiziert ihre IP-Adressen und definiert dann das Gateway payroll_gw als payroll_gw_interface1/2 als seine beiden Schnittstellen aufweisend. Das Codefragment weist außerdem den Schnittstellen die Rollengruppe Gateway zu.

```
HOST {

        payroll_gw_interface1 = [111.222.26.226 ] : gateway

        payroll_gw_interface2 = [111.222.24.210 ] : gateway

}

GATEWAYS {

        payroll_gw = { payroll_gw_interface1, payroll_gw_interface2}

}
```

[0087] Zonen werden mittels der folgenden Anweisung definiert:
<zone-name> : {<gtwy-interface-name1>, <gtwy-interface-name2> ...}

[0088] Das folgende Codefragment definiert zunächst payroll_zone und corp_zone (Teile des Intranet manhattan_office) als Hostgruppen, spezifiziert ihre IP-Bereiche und definiert dann Teile der Netzwerktopologie durch Spezifizieren von payroll_zone zur Verbindung mit payroll_gw durch Mittel der payroll_gw_interfacel, und die zweite Schnittstelle von payroll_gw zur Verbindung mit corp_zone.

```
            HOST-GROUPS {

            manhattan_office = [111.222.0.0-111.222.255.255] : intranet_machine

            payroll_zone     = [111.222.26.0-111.222.26.255] : payroll_machine

            corp_zone        =[111.222.24.0-111.222.24.255] :
non_payroll_machine

            }


                ZONES {

                payroll_zone  = {payroll_gw_interface1}

                corp_zone     = { payroll_gw_interface2, ... }

                }
```

[0089] Es versteht sich, daß die hier gezeigten und beschriebenen Ausführungsformen und Varianten lediglich die Prinzipien der vorliegenden Erfindung veranschaulichen, und daß Fachleute verschiedene Modifikationen implementieren können.

Patentansprüche

1. Verfahren zum Analysieren mindestens eines Gateway (**120**) in einem Netzwerk (**100**), wobei das mindestens eine Gateway eine Paketfilterungskonfigurationsdatei (**125**) aufweist, die mehrere Regeln enthält, wobei das Netzwerk mehrere Adressen aufweist, wobei das Verfahren die folgenden Schritte umfaßt:
Erzeugen eines Gateway-Zonen-Graphen, der das Netzwerk modelliert, wobei der Gateway-Zonen-Graph mindestens einen dem mindestens einen Gateway entsprechenden Gateway-Knoten und mindestens zwei Zonenknoten aufweist, wobei das mindestens eine Gateway eine Paketfilterungsmaschine ist und jeder der Zonenknoten einer partitionierten Ansammlung der Adressen entspricht, die durch das mindestens eine Gateway erzeugt wird;
Empfangen (**410**) einer Anfrage, die anfragt, ob ein oder mehrere gegebene Dienste zwischen mindestens einer Quellenadresse und mindestens einer Zieladresse erlaubt sind; und
Auswerten (**420**, **430**, **440**, **450**) der Anfrage im Hinblick auf die Regeln, die mit jedem Gateway-Knoten in dem Gateway-Zonen-Graphen, der zwischen der mindestens einen Quellenadresse und der mindestens einen Zieladresse angetroffen wird, assoziiert sind, wobei die Regeln aus der jeweiligen Paketfilterungskonfigurationsdatei abgeleitet werden.

2. Verfahren nach Anspruch 1, wobei die Regeln als auf Regeln basierende Objekte ausgedrückt werden.

3. Verfahren nach Anspruch 1, wobei der Gateway-Zonen-Graph aus einer Netzwerktopologiedatei (**210**) abgeleitet wird.

4. Verfahren nach Anspruch 1, wobei die Anfrage eine Wildcard für den Dienst, die Quellenadresse und/oder die Zieladresse enthält.

5. Verfahren nach Anspruch 1, ferner mit dem Schritt des Bestimmens eines Teils des einen bzw. der mehreren gegebenen Dienste, die zwischen mindestens einer Quellenadresse und mindestens einer Zieladresse erlaubt sind.

6. Verfahren nach Anspruch 1, ferner mit dem Schritt des Transformierens der Paketfilterungskonfigurationsdateien in eine Tabelle logischer Regeln, die während des Auswertungsschritts verarbeitet werden.

7. Verfahren nach Anspruch 1, wobei die Anfrage die Parameter-Quellen-Host-Gruppe, die Ziel-Host-Gruppe und Dienst-Host-Gruppe umfaßt.

8. Verfahren nach Anspruch 1, wobei die Anfrage eine Stelle spezifiziert, an der Pakete in das Netzwerk eingefügt werden sollen, die von einer Quellenadresse verschieden ist.

9. Vorrichtung zum Analysieren mindestens eines Gateway (**120**) in einem Netzwerk (**100**), wobei das mindestens eine Gateway eine Paketfilterungskonfigurationsdatei (**125**) aufweist, die mehrere Regeln enthält, wobei das Netzwerk mehrere Adressen aufweist, wobei die Vorrichtung folgendes umfaßt:
einen Speicher zum Speichern von computerlesbarem Code; und
einen wirksam an den Speicher angekoppelten Prozessor, wobei der Prozessor für folgendes konfiguriert ist:
Erzeugen eines Gateway-Zonen-Graphen, der das Netzwerk modelliert, wobei der Gateway-Zonen-Graph mindestens einen dem mindestens einen Gateway entsprechenden Gateway-Knoten und mindestens zwei Zonenknoten aufweist, wobei das mindestens eine Gateway eine Paketfilterungsmaschine ist und jeder der Zonenknoten einer partitionierten Ansammlung der Adressen entspricht, die durch das mindestens eine Gateway erzeugt wird;
Empfangen (**410**) einer Anfrage, die anfragt, ob ein oder mehrere gegebene Dienste zwischen mindestens einer Quellenadresse und mindestens einer Zieladresse erlaubt sind; und
Auswerten (**420**, **430**, **440**, **450**) der Anfrage im Hinblick auf die Regeln, die mit jedem Gateway-Knoten in dem Gateway-Zonen-Graphen, der zwischen der mindestens' einen Quellenadresse und der mindestens einen Zieladresse angetroffen wird, assoziiert sind, wobei die Regeln aus der jeweiligen Paketfilterungskonfigurationsdatei abgeleitet werden.

10. Vorrichtung nach Anspruch 9, wobei die Regeln als auf Regeln basierende Objekte ausgedrückt werden.

11. Vorrichtung nach Anspruch 9, wobei der Gateway-Zonen-Graph aus einer Netzwerktopologiedatei (**210**) abgeleitet wird.

12. Vorrichtung nach Anspruch 9, wobei die Anfrage eine Wildcard für den Dienst, die Quellenadresse und/oder die Zieladresse enthält.

13. Vorrichtung nach Anspruch 9, ferner mit dem Schritt des Bestimmens eines Teils des einen bzw. der mehreren gegebenen Dienste, die zwischen mindestens einer Quellenadresse und mindestens einer Zieladresse erlaubt sind.

14. Vorrichtung nach Anspruch 9, ferner mit dem Schritt des Transformierens der Paketfilterungskonfigurationsdateien in eine Tabelle logischer Regeln, die während des Auswertungsschritts verarbeitet werden.

15. Vorrichtung nach Anspruch 9, wobei die Anfrage die Parameter-Quellen-Host-Gruppe, die Ziel-Host-Gruppe und Dienst-Host-Gruppe umfaßt.

16. Vorrichtung nach Anspruch 9, wobei die Anfrage eine Stelle spezifiziert, an der Pakete in das Netzwerk eingefügt werden sollen, die von einer Quellenadresse verschieden ist.

17. Computerlesbares Medium mit darauf realisierten computerlesbaren Programmcodemitteln zum Bewirken, daß ein Computer die Schritte eines Verfahrens nach einem der Ansprüche 1 bis 8 ausführt.

Es folgen 6 Blatt Zeichnungen

Anhängende Zeichnungen

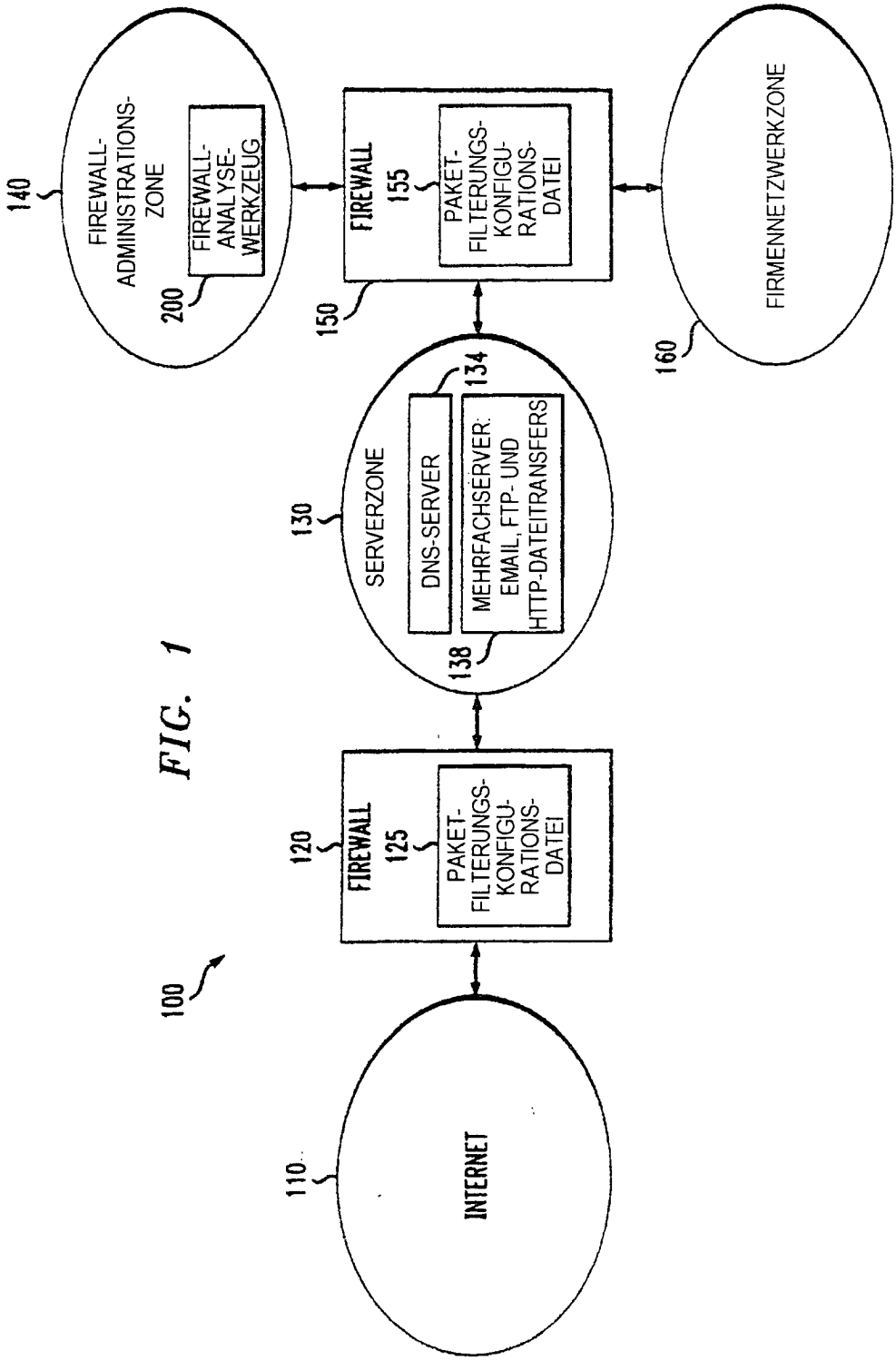


FIG. 1

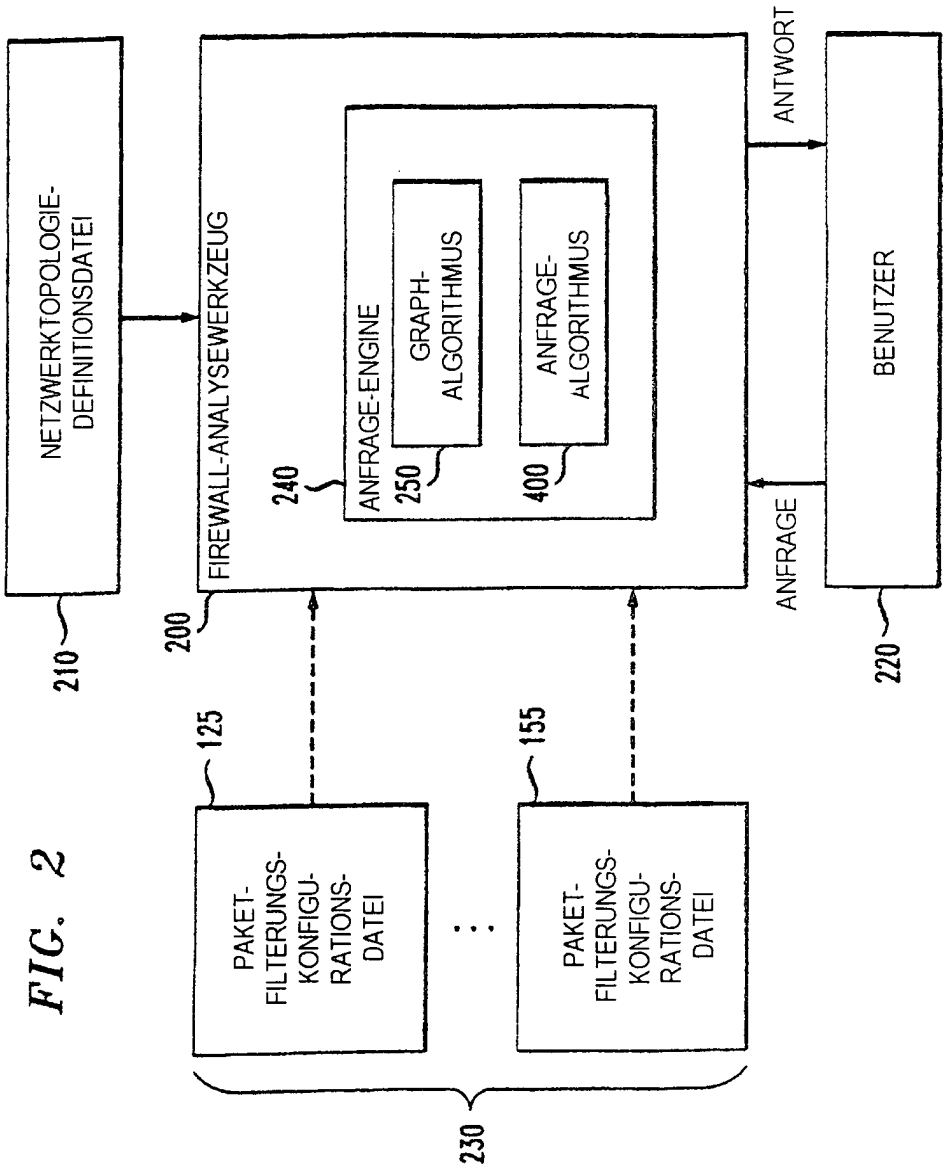

NETZWERKTOPOLOGIE-DEFINITIONSDATEI
210
FIREWALL-ANALYSEWERKZEUG
200
240
ANFRAGE-ENGINE
250
GRAPH-ALGORITHMUS
400
ANFRAGE-ALGORITHMUS
ANTWORT
ANFRAGE
BENUTZER
220
125
155
PAKET-FILTERUNGS-KONFIGU-RATIONS-DATEI
PAKET-FILTERUNGS-KONFIGU-RATIONS-DATEI
230

FIG. 2

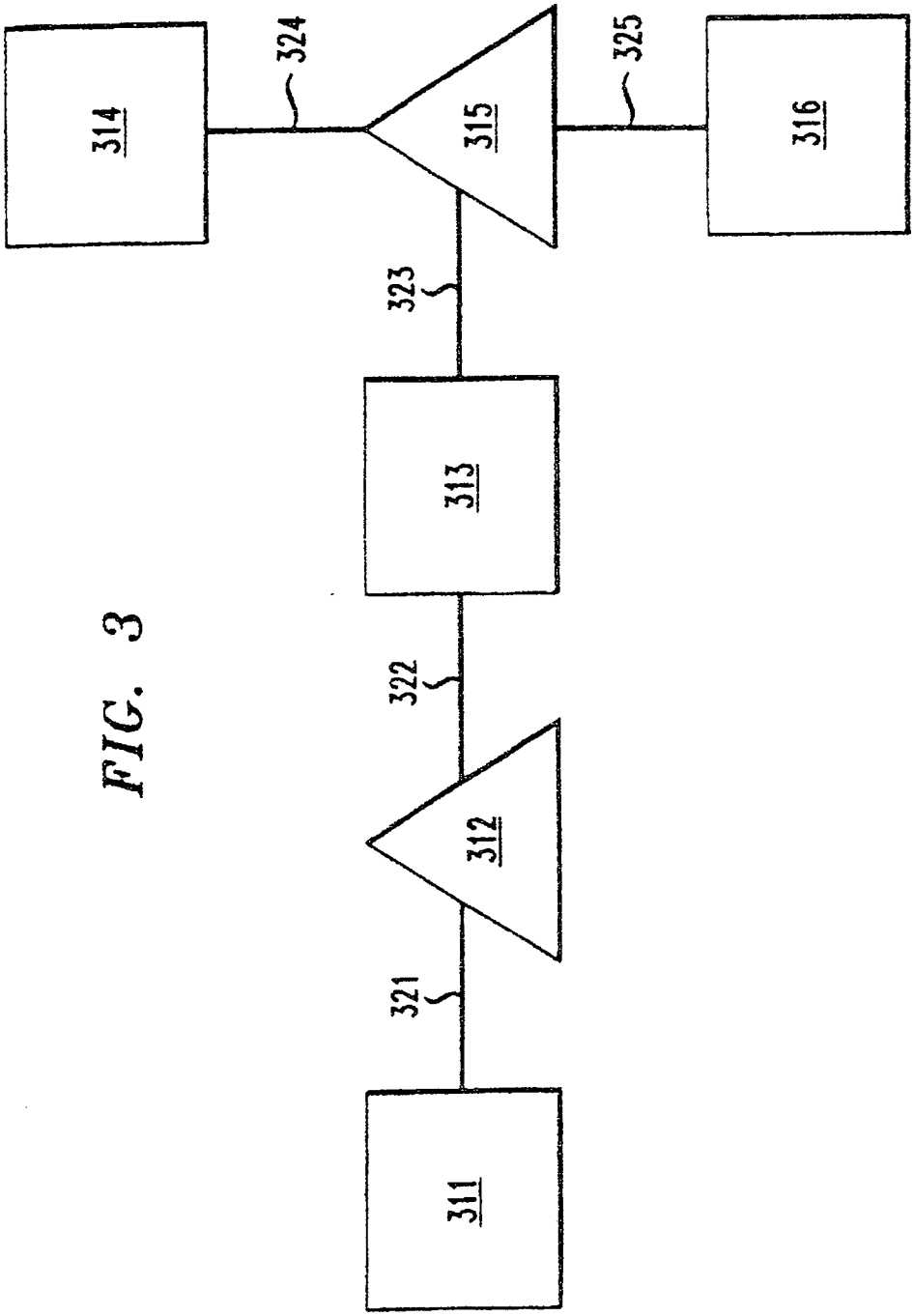


FIG. 3

FIG. 4

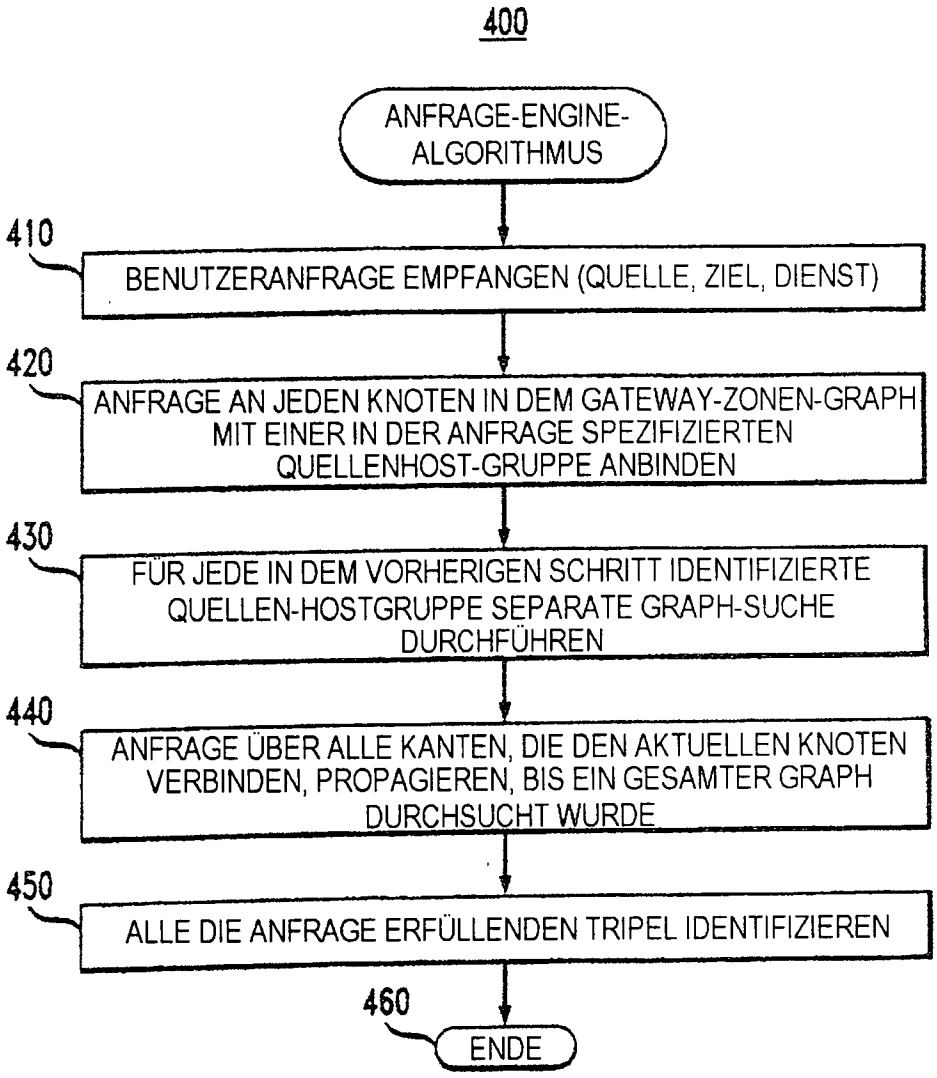

400
ANFRAGE-ENGINE-ALGORITHMUS
410
BENUTZERANFRAGE EMPFANGEN (QUELLE, ZIEL, DIENST)
420
ANFRAGE AN JEDEN KNOTEN IN DEM GATEWAY-ZONEN-GRAPH MIT EINER IN DER ANFRAGE SPEZIFIZIERTEN QUELLENHOST-GRUPPE ANBINDEN
430
FÜR JEDE IN DEM VORHERIGEN SCHRITT IDENTIFIZIERTE QUELLEN-HOSTGRUPPE SEPARATE GRAPH-SUCHE DURCHFÜHREN
440
ANFRAGE ÜBER ALLE KANTEN, DIE DEN AKTUELLEN KNOTEN VERBINDEN, PROPAGIEREN, BIS EIN GESAMTER GRAPH DURCHSUCHT WURDE
450
ALLE DIE ANFRAGE ERFÜLLENDEN TRIPEL IDENTIFIZIEREN
460
ENDE

FIG. 5

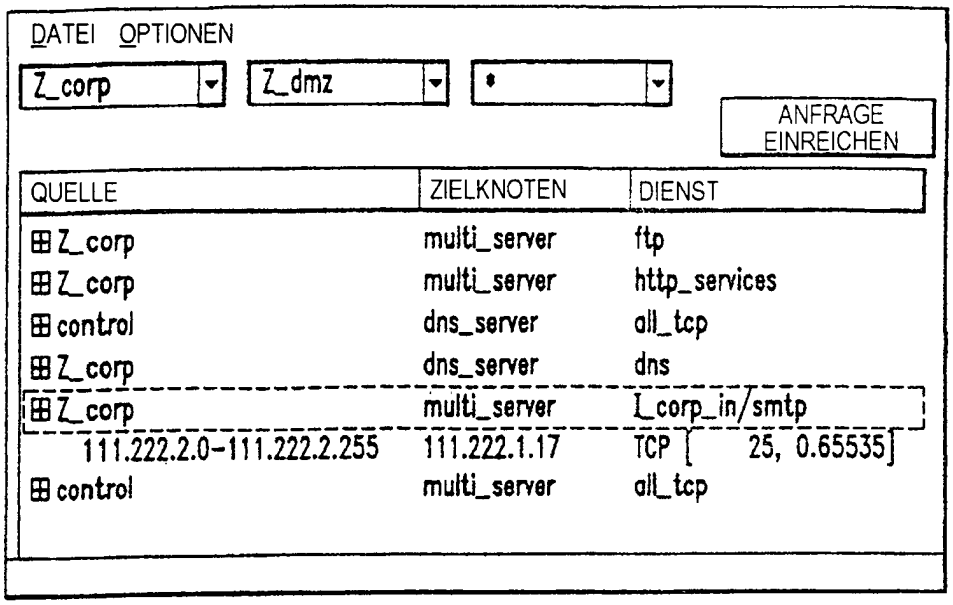

DATEI OPTIONEN
Z_corp
Z_dmz
*
ANFRAGE EINREICHEN
QUELLE
ZIELKNOTEN
DIENST
Z_corp
multi_server
ftp
Z_corp
multi_server
http_services
control
dns_server
all_tcp
Z_corp
dns_server
dns
Z_corp
multi_server
L_corp_in/smtp
111.222.2.0-111.222.2.255
111.222.1.17
TCP [25, 0.65535]
control
multi_server
all_tcp

FIG. 6

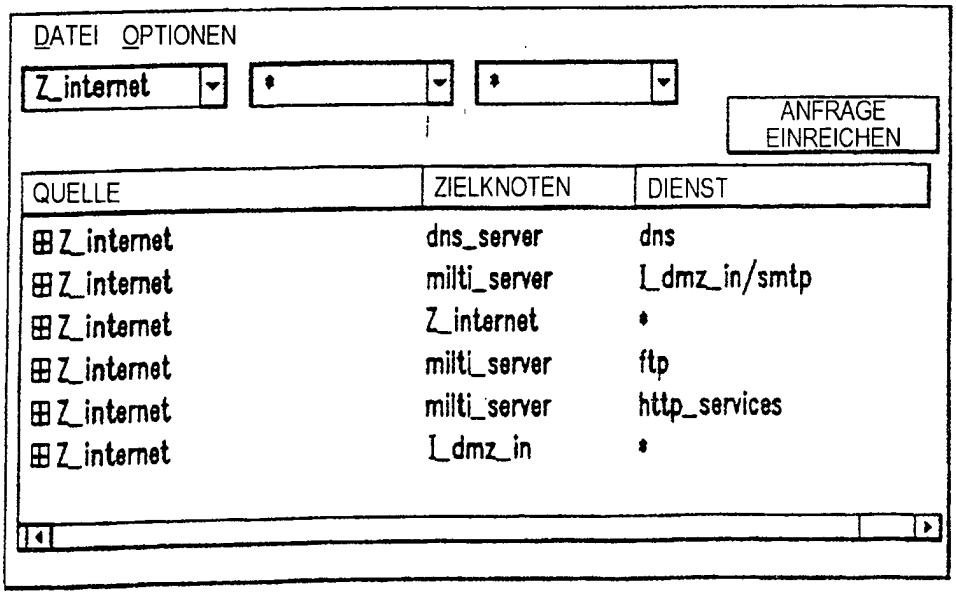
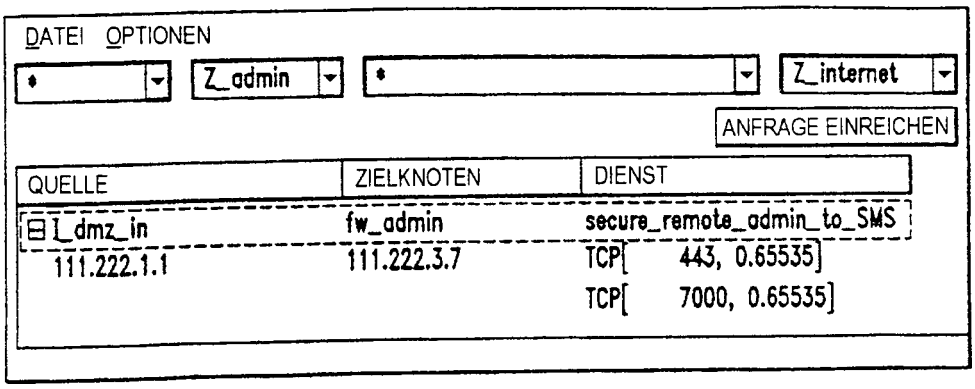

DATEI OPTIONEN
Z_internet
*
*
ANFRAGE EINREICHEN
QUELLE
ZIELKNOTEN
DIENST
Z_internet
dns_server
dns
Z_internet
milti_server
L_dmz_in/smtp
Z_internet
Z_internet
*
Z_internet
milti_server
ftp
Z_internet
milti_server
http_services
Z_internet
L_dmz_in
*

FIG. 7


DATEI OPTIONEN
Z_admin
Z_internet
ANFRAGE EINREICHEN
QUELLE
ZIELKNOTEN
DIENST
l_dmz_in
fw_admin
secure_remote_admin_to_SMS
111.222.1.1
111.222.3.7
TCP[443, 0.65535]
TCP[7000, 0.65535]