(19) Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 197 40 547 B4** 2006.03.30

(12) **Patentschrift**

(21) Aktenzeichen: **197 40 547.9**
(22) Anmeldetag: **15.09.1997**
(43) Offenlegungstag: **16.04.1998**
(45) Veröffentlichungstag der Patenterteilung: **30.03.2006**

(51) Int Cl.[8]: ***H04L 12/22*** (2006.01)
G06F 12/14 (2006.01)

(30) Unionspriorität:
08/713,424 13.09.1996 US

(73) Patentinhaber:
Secure Computing Corp., Roseville, Minn., US

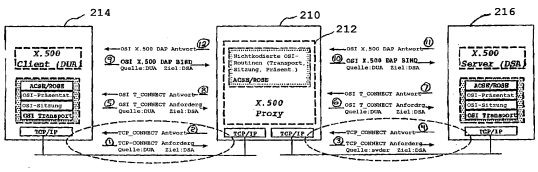
(74) Vertreter:
Eisenführ, Speiser & Partner, 28195 Bremen

(72) Erfinder:
Green, Michael W., Shoreview, Minn., US; Kruse, Ricky Ronald, Ham Lake, Minn., US

(56) Für die Beurteilung der Patentfähigkeit in Betracht gezogene Druckschriften:
US 55 50 984 A
EP 02 03 424 A2
Internet Request for Comment-RFC 1006 Network Working Group, Northern Research and Technology Center, Mai 1987;
BELLOVIN, S.M. und CHESWICK, W.R.: Network Firewalls. In: IEEE Communications Magazine, Sept. 1994, S. 50-57;
SHARP, R.L., et al.: Network Security in a Heterogeneous Environment. In: AT&T Technical Journal, Sep./Oct. 1994, S. 52-60;

(54) Bezeichnung: **Vorrichtung und Verfahren zum Sicherstellen sicherer Kommunikation zwischen einer anfordernden Entität und einer bedienenden Entität**

(57) Hauptanspruch: Vorrichtung zum Sicherstellen sicherer Kommunikation zwischen einer anfordernden Entität und einer bedienenden Entität, mit:
– Mitteln zum Annehmen einer Verbindungsanforderung der anfordernden Entität, welche eine Verbindung zu der bedienenden Entität anfordert,
– Mitteln zum Aufbauen einer auf einer Vielzahl von unterschiedlichen Schichten arbeitenden Sitzungs-Verbindung mit der anfordernden Entität,
– Mitteln zum Überprüfen der Kommunikation von der anfordernden Entität hinsichtlich der Übereinstimmung mit einem selektierten Kommunikationsprotokoll auf den oberen der Vielzahl von unterschiedlichen Schichten,
– Mitteln zum Aufbauen einer Sitzungs-Verbindung mit der bedienenden Entität, falls durch die Mittel zum Überprüfen der Kommunikation Übereinstimmung mit dem selektierten Kommunikationsprotokoll festgestellt wurde, und
– Mitteln zum Vermitteln der Kommunikation zwischen der anfordernden Entität und der bedienenden Entität auf der Transportschicht, wenn beide Verbindungen aufgebaut sind,
wobei die Vorrichtung so ausgebildet ist, dass sie für die beiden Entitäten transparent erscheint.



Beschreibung

[0001] Die Erfindung betrifft eine Vorrichtung und ein Verfahren zum Sicherstellen sicherer Kommunikation zwischen einer anfordernden Entität und einer bedienenden Entität.

Stand der Technik

[0002] Die EP 0 203 424 A2 offenbart ein Verfahren und eine Schaltungsanordnung zum Überprüfen der Berechtigung des Zugangs zu einem Signalverarbeitungssystem. In diesem Stand der Technik wird die Verwendung eines bestimmten Schlüsselwortes für jede periphere Einrichtung offenbart, um den Zugang zu autorisieren.

[0003] Netzwerke verbinden mehrere Computer untereinander and erlauben ihnen, Daten über Kommunikationsleitungen auszutauschen. Verschiedene Standards, die festlegen, wie ein solcher Datenaustausch stattfindet, wurden entwickelt and implementiert, um sicherzustellen, dass Computer and Computerprogramme, welche die gleichen Protokolle verwenden, erfolgreich Daten austauschen können. Eines der mit der Fähigkeit zum Austauschen von Daten einhergehenden Probleme ist, sicherzustellen, dass eine Anforderungs-Entität, wie ein Benutzer in einem Netzwerk, autorisiert ist, auf Daten in einer Server-Entität, wie einem anderen Computer, zuzugreifen.

[0004] Firewalls sind Vorrichtungen wie Programme oder separate Computer-Systeme, welche eingesetzt werden, um die Sicherheitsprobleme anzusprechen, die einhergehen mit dem Verbinden eines einerseits privaten Netzwerkes wie einem lokalen Netz (Local-Area-Network), welches Computer in einem Büro verbindet, mit einem "Internet", wobei die Datenübertragungen offen sind zum Abhören und das Potential für "feindliche" Außenseiter vorhanden ist, Netzwerkdienste zu unterbrechen oder zu manipulieren oder Systeme anzugreifen, die in dem privaten Netzwerk vorhanden sind.

[0005] Es gibt eine Anzahl unterschiedlicher Klassen von Firewalls, die jeweils ausgebildet sind, um unterschiedlichen Arten von Sicherheitsbedürfnissen zu entsprechen. Trotz der unterschiedlichen Ansätze führen alle Firewalls eine Funktion aus, die als "Übermitteln" bekannt ist, wobei Protokoll-Daten-Einheiten (PDUs) durch den Firewall von einer Sende-Entität empfangen werden und zu einer Empfangs-Entität weitergeleitet werden, möglicherweise mit einigen Modifikationen an der ursprünglichen PDU. Da Firewalls ausgebildet sind, um eine Sicherheitspolitik durchzusetzen, müssen einige Informationen oder ein Kontext aus den PDUs extrahiert und einem Regelsatz unterworfen werden. Basierend auf dem Ergebnis der Regelprüfung führt der Firewall einen Vorgang aus; die PDU wird entweder übermittelt, modifiziert und übermittelt oder in verschiedener Weise zurückgewiesen. Der genaue Vorgang wird durch den Designer des Firewalls gewählt, um das Verhalten des Systems so zu beeinflussen, daß die Sicherheitspolitik erfüllt wird. Der Vorgang ist natürlich den Beschränkungen des Protokolls unterworfen, welches der Firewall unterstützt.

[0006] Eine von Anwendungs-Entitäten zum Datenaustausch verwendete Gruppe von Protokollen wird als Open Systems Interconnect (OSI) bezeichnet. OSI-Anwendungen sind aufgebaut auf der Basis eines 7-Schichten-Modells. Oben beginnend wird die Schicht 7 als die Anwendungsschicht bezeichnet, Schicht 6 ist die Präsentationsschicht, 5 die Sitzung, 4 ist der Transport, 3 ist das Netzwerk, 2 ist die Datenverbindung und 1 ist physikalisch. Beginnend am Boden und aufwärts gehend handhabt die physikalische Schicht die Übertragung der Bits über ein Kommunikationsmedium wie eine Telefonleitung. Die Datenverbindungsschicht sammelt die Bits in einer Gruppe von Bits, die als Rahmen bezeichnet wird. Die Netzwerkschicht routet die Rahmen zwischen Knoten in dem Netzwerk, nennt aber die Rahmen Datenpakete. Diese drei unteren Schichten werden durch Kommunikationsgeräte einschließlich Schaltern in dem Netzwerk implementiert. Die Transportschicht behandelt diese Pakete wie Mitteilungen und ist die untere Schicht in einem Computer. Sie befindet sich zwischen der inneren Routing- oder Sitzungs-Schicht in dem Computer und dem Netzwerk. Die Sitzungs-Schicht liefert die Verbindung, welche die unterschiedlichen Datenströme verbindet, wie die Audio- und Videoteile einer Telekonferenz-Anwendung. Die nächsthöhere Schicht ist die Präsentationsschicht, welche sich mit dem Format der ausgetauschten Daten befaßt. Schließlich ist die Anwendungsschicht die oberste Schicht und kann, mit Begriffen des Internet, als das File Transfer Protocol (FTP), einen der zum Datenaustausch im Dateiformat im Internet verwendeten Mechanismen, vorgesehen sein.

[0007] Jede Schicht N stellt Dienste für die Schicht oberhalb (Schicht N + 1) bereit und benötigt Dienste der Schicht unterhalb (N – 1). Firewalls arbeiten typisch als ein Protokoll-Peer in einer bestimmten Schicht (z.B. der Transportschicht, N = 4) und übertragen die Protokollinformationen und Daten (PDUs) in der Schicht oder leiten sie weiter. Ein in der Internet-Protokollgruppe arbeitendes Schicht-4-Relais arbeitet in der Transaction Control Protocol (TCP) Protokollschicht und leitet TCP-Segmente (Anwendungsdaten) zwischen kommunizierenden Anwendungs-Entitäten weiter und wird als Protokoll-Peer in der TCP-Schicht betrachtet.

[0008] Für Anwendungen, die ausgebildet sind, um in der Internet-Gruppe zu laufen, ist die Schicht 7 die

Anwendungsschicht, wenn sie auf das OSI-Modell übertragen werden, wobei die Schichten 5 und 6 durch das TCP implizit bereitgestellt werden. Somit können Internet-Firewalls anwendungsspezifische Sicherheitsprüfungen ausführen durch einfaches Überwachen der durch die Transportschicht transportierten PDUs. Wenn eine Sicherheitsverletzung erfaßt wird, kann eine bedeutsame, anwendungsspezifische Protokollantwort aus der Kontextinformation erzeugt werden, die während der Sitzung aufgezeichnet wurde. Internet-Firewalls reagieren auf Sicherheitsverletzungen durch Erzeugen einer Reaktion in der Schicht N oder N + 1, wobei N die Schicht ist, in welcher der Firewall als ein Peer (Relais) wirkt. Ein als Schicht-4-Relais wirkender FTP-Firewall kann einen Verbindungsversuch von einem unerwünschten Clienten zurückweisen durch Ausführen eines TCP-Schließen (Schicht-4-Vorgang) oder kann einen Versuch zum Ablegen oder Speichern einer Datei durch eine FTP-Fehlerantwort (Schicht-7-Vorgang) zurückweisen. Es ist anzumerken, daß die Vorgänge der Schichten 5 und 6 implizit durch das TCP bereitgestellt werden.

[0009] Da Internet-Anwendungsprotokolle dazu neigen, textbasiert zu sein, können die Datenerfassungs- und Antwort-Mechanismen von relativ einfachen Parsern und Kodierern aufgebaut werden und benötigen keine große Mengen von zu unterstützender Zustandsinformation. Internet-Firewalls können den Anwendungs-Kontext identifizieren durch Prüfen der Ziel-TCP-Anschlußnummer, an die sich die Client-Internet-Anwendung anzuschließen versucht. Ein Internet-Firewall muß nicht jede Anwendungsschicht-PDU prüfen, um die Art des angeforderten Dienstes zu bestimmen. Ein Internet-Firewall kann z.B. zwischen einer Anforderung des Simple Mail-Transfer-Protocol (SMTP) Dienstes für E-mail von demjenigen des FTP-basierten lediglich anhand der Anschlußnummer unterscheiden. Somit kann ein Internet-Firewall einfach durch Annehmen von Anforderungen, die an den SMTP-Anschluß gerichtet sind, aber Zurückweisen von Anforderungen an den FTP-Anschluß konfiguriert sein, um SMTP zuzulassen, aber FTP zu verweigern.

[0010] Für OSI-Anwendungen ist der Aufbau komplexer. Es wird erwartet, daß OSI-Anwendungen strikt dem 7-Schichten-Modell (keine impliziten Schichten) entsprechen und von den OSI-Sitzungs- und Präsentations-Schichten Gebrauch machen. Jede dieser Schichten führt zusätzliche Zustände in die Sitzung ein und sie wirken als Protokoll-Entitäten mit eigenen Rechten. Ein Fehler in einer höheren Schicht muß über und zu in niedrigeren Schichten vorgesehenen Diensten transportiert werden.

[0011] Während einer Verbindungs-Aufbauphase versuchen die Transportschicht, die Sitzungsschicht und die Präsentationsschicht eine Zuordnung zu ihrem entsprechenden Protokoll-Peer zu bilden und Sitzungs-Parameter auszuhandeln. Die Anwendungsschicht wird tatsächlich aus mehreren Application Service Elements (ASEs) gebildet, von denen jedes Unterstützung für einen anderen Satz zugehöriger Dienste und Protokolle bereitstellt, und sie können zum Bilden von "Unter-Schichten" der Anwendungsschicht angeordnet sein. Gemeinsame ASEs beinhalten das Association Control Service Element (ACSE) und das Remote Operations Service Element (ROSE) und die Kernelemente werden in den meisten OSI-Anwendungen gefunden. ACSE wird verwendet, um Zuordnungen (Verbindungen) aufzubauen und abzubauen und ROSE wird verwendet, um Anforderungen in einer einheitlichen Weise zu transportieren. Das Identifizieren von Benutzer-Ausweisen und Authentifizierungs-Informationen werden mit dem gewählten Protokoll (X.500, X.400 oder FTAM, um einige zu nennen) ausgetauscht, wie der Unterscheidungsname und ein Password oder eine kryptographische Signatur des Benutzers. Der Vorgang des Ausbildens einer OSI-Anwendungsschicht-Zuordnung ist bekannt als Binding.

[0012] Aus dem Aufsatz „Network Firewalls" von Steven M. Bellovin und William R. Cheswick auf den Seiten 50 bis 57 des „IEEE Communication Magazine" von September 1994 ist es bekannt, dass Firewalls gleichzeitig auf verschiedenen OSI-Schichten arbeiten können.

[0013] Sobald eine Anwendung an ihren Protokoll-Peer gebunden ist, wird die Anwendung als „in einer Sitzung (in session)" bezeichnet und die Ausführung tritt in einen Bereitschaftszustand ein, wobei die Anwendungsschicht-PDUs ausgetauscht und verarbeitet werden, bis einer der Anwendungs-Peers das Schließen der Sitzung auslöst. Eine ordentliche Freigabe der Zuordnung wird als Unbindung bezeichnet.

[0014] Sämtliche OSI-Anwendungs-PDUs werden binär übertragen, Anwendungen und Präsentationsschichten werden typisch in einem ASN.1 genannten Format binär kodiert; die Sitzung und die Transportschichten fragmentieren die Information.

[0015] OSI-Anwendungen werden ausgebildet, um in einer Umgebung zu arbeiten, in welcher die OSI-Transportschicht verwendet wird. Ein Verfahren wurde entwickelt, um OSI-Transportdienste zu dem Internet-Stack umzuleiten. Dieses Verfahren ist beschrieben in Internet Request for Comment-RFC-1006, welche auf vielen Servern im Internet in einer Datei mit dem Namen RFC-1006.TXT gespeichert ist. Bei dem RFC-1006-Verfahren wird die OSI-Transportklasse 0 (TPO) selektiert und TP0-PDUs (T-PDU) werden mit einem kurzen Vier-Oktett-Header eingefaßt, der als ein T-Paket (TPKT) bezeichnet wird. Die RFC schlägt vor, daß

Server, welche dieses Verfahren implementieren, an dem TCP-Anschluß 102 antworten, welcher für OSI über TCP reserviert wurde, der tatsächlich gewählte Anschluß bleibt jedoch dem Ermessen eines Administrators oder Implementierers vorbehalten.

[0016] Ein IP-transparentes Relais fragt beide Leitungen nach bestimmten, zu sendenden IP-Datenrahmen ab. Die IP-transparente Bridge enthält eine Liste von Netzwerkadressen, die jeder Leitung zugeordnet sind, so daß, wenn einer dieser Rahmen empfangen wird, eine Liste überprüft wird und die IP-Bridge ihn "ergreift" und ihn in dem anderen Netzwerk ablegt. Das Problem bei dieser einfachen IP-transparenten Bridge-Lösung ist, daß sie nur Informationen filtern kann, welche auf den IP-Adressen basieren, die in den Daten enthalten sind. Eine IP-Adressen-Vortäuschung ist sehr leicht, daher ist diese Lösung alleine nicht sicher genug für die Bedürfnisse der Absicherung einer OSI-Anwendung.

[0017] Die transparente Relais-Funktionalität muß wieder höher hinaufbewegt werden, zu diesem Zeitpunkt zu der OSI-Transport-Schicht. Die Transportschicht über IP ist TCP. Eine TCP-transparente Relais-Lösung wartet auf Daten an bestimmten TCP-Anschlüssen. Die TCP-Bridge filtert, basierend auf einem bestimmten TCP-Anschluß, welcher allgemein einer einzelnen Anwendung zugeordnet ist. Das Problem ist jedoch, daß die OSI-Anwendungen nicht darauf ausgerichtet sind, mit jedem bestimmten TCP-Anschluß zu arbeiten. Daher muß die Lösung weiterhin die Daten prüfen, die hindurchlaufen, um sicherzustellen, daß sie mit dem Inhalt übereinstimmen, der von einer Anwendung erwartet wird, die durch ihn kommuniziert. Um die Daten zu verifizieren, muß die TCP-Bridge aber selbst auf die Verbindungsanforderung antworten. Sobald die Antwort empfangen ist, beginnt der Absender, Anwendungsdaten zu der TCP-Bridge zu senden. Die TCP-Bridge prüft dann die Daten und entscheidet, ob sie mit der Form übereinstimmen, die für diese Anwendung erwartet wird. Ist dies der Fall, baut die TCP-Bridge eine unabhängige Sitzung mit dem "realen" Zielgerät auf und leitet die Daten des Senders zu ihm weiter. Das Hinzufügen dieser zusätzlichen Verarbeitung zum Akzeptieren einer Verbindungsanforderung und weiteres Prüfen der Daten beinhaltet die Funktionalität über eine einfache transparente Bridge hinaus. Allgemein ist diese vollständige Kombination von Funktionalität das, was als eine "Proxy"-Lösung bezeichnet wird.

[0018] Internet-Firewalls nähern sich der Frage, wie eine sichere Unterstützung für OSI-Protokolle bereitzustellen ist, unter Verwendung von zwei hauptsächlichen Lösungsansätzen, Anwendungs-Gateways oder Proxies.

[0019] Anwendungs-Gateways sind eine besondere Form von Firewall, wobei der Firewall PDUs in der Anwendungsschicht annimmt und verarbeitet und in dem Netzwerk als ein Protokoll-Peer für den Client und den Server erscheint. Für OSI-Protokolle werden durch den Anwendungs-Gateway alle sieben Schichten verarbeitet und getrennte Anwendungsschicht-Zuordnungen werden zwischen dem Firewall und dem Client und zwischen dem Firewall und dem Server aufrechterhalten. Somit wird der Firewall als "sichtbar" für den Client und den Server bezeichnet, da er eine direkt angesprochene Anwendungs-Entität ist; die Client-Anwendung muß somit darauf achten, wie der Firewall angesprochen wird, um den Server zu erreichen.

[0020] Anwendungs-Gateways sind in der Lage, den vollständigen Kontext der verarbeiteten Information durch ihren Betrieb in der höchsten Schicht aufzunehmen, was ihnen einen deutlichen Vorteil vor traditionellen N-Schicht-Proxies (nachfolgend erläutert) beim Versuch, eine Protokoll-spezifische Sicherheitspolitik durchzusetzen, gibt.

[0021] Für die Internet-Anwendungen bieten Anwendungs-Gateways die höchste Sicherheit und aufgrund der Einfachheit der Protokolle ist es möglich, sie zu implementieren. OSI-Protokolle sind jedoch sehr komplex und betreffen einige Schichten oberhalb der TCP-Schicht. Da der Gateway ein Protokoll-Peer ist, muß er weiterhin auch alle notwendigen Protokollelemente unterstützen, die erforderlich sind, damit er ein tatsächlicher Server für die Anwendung ist.

[0022] Aufgrund der Komplexität der erforderlichen Serviceelemente und der oberen Schichten des OSI-Stacks neigen Anwendungsschicht-Gateways für OSI-Dienste dazu, Implementationen der Anwendungs-Server selbst manchmal mit begrenzter Funktionalität zu sein. Ein Anwendungs-Gateway für ein von der International Telecommunication Union veröffentlichtes, mit X.500 bezeichnetes Kommunikationsprotokoll, nimmt gewöhnlich die Form eines X.500 Directory System Agent (DSA) an, welcher die Server-Komponente von X.500 ist. Der DSA wird modifiziert, um die Sicherheitspolitik-Entscheidungen bei der Unterstützung der Firewall-Funktionalität zu unterstützen.

[0023] Während diese Anordnung das Potential für eine sehr gute Sicherheit bietet, ist die Leistungsfähigkeit ein Problem, da der Firewall die Funktionalität des Servers implementieren oder simulieren muß, welche oft komplexe Berechnungen betrifft, Datenmanipulationen und eine beträchtlichen Menge gespeicherter Statusinformationen. Der Firewall kann große Datenmengen zwischenspeichern müssen, bevor er in der Lage ist, die Daten zu der anderen unabhängigen Anwendungszuordnung zu übermitteln. Die Komplexität dieser Lösung macht es ebenfalls

sehr schwierig, die Korrektheit der Implementation zu prüfen und die resultierende Implementation nach Sicherheitslücken und Schwachpunkten zu analysieren.

[0024] Der zweite wesentliche Ansatz betrifft filternde N-Schicht-Relais (Proxies). Ein N-Schicht-Relais wirkt als eine Bridge, welche in einem Netzwerk gesendete PDUs aufnimmt und sie in ein anderes Netzwerk zurücksendet. Diese Vorrichtungen werden als "transparent" bezeichnet, da keine End-Stations-Anwendungs-Entität das Relais wahrnimmt.

[0025] Um als Proxy bezeichnet zu werden, muß ein N-Schicht-Relais eine Firewall-Funktion bei der Unterstützung einer Sicherheitspolitik ausführen. Proxies arbeiten stets unterhalb der Anwendungsschicht und Filter-PDUs, die auf in dieser Schicht sichtbaren Attributen basieren.

[0026] Wenn ein Proxy in einer als eine MAC-Schicht bezeichneten Daten-Verbindungsschicht arbeiten soll, erfaßt er Ethernet-Rahmen und prüft die Adressen in dem MAC-Header und filtert den Nutzteil (IP-Datagramme) zum Bestimmen der Internet-Protokoll-(IP)-Adressen. Das Filtern einer höheren Schicht ist unausführbar, da Daten zwischengespeichert und neu zusammengesetzt werden müssen, um ausreichend Kontext zu erhalten und die Semantik von TCP ist derart, daß nur eine begrenzte Anzahl von Rahmen zwischengespeichert und geprüft werden kann, bevor sie gesendet werden müssen, um weitere zu empfangen. Wenn nur ein teilweiser Sicherheits-Kontext bestimmt wurde, wenn der Puffer-Schwellwert erreicht wurde, müssen die Daten entweder verworfen oder ohne vollständige Validierung gesendet werden, in jedem Fall eine unannehmbare Alternative für eine OSI-Anwendung.

[0027] IP-Schicht-Proxies haben im wesentlichen die gleichen Kennzeichen wie der MAC-Proxy hinsichtlich der Politik und den Beschränkungen. IP-Proxies fassen IP-Datagramme zusammen und können das IP, den Header und gewöhnlich ebenso das TCP filtern. Das IP-Proxy-Firewall-Verhalten ist bei den meisten modernen, kommerziellen Routern verfügbar.

[0028] TCP-Proxies können IP-Adressen, TCP-Anschlußnummern und andere in der TCP-Schicht sichtbare Attribute filtern und TCP-Segmente von einem Netzwerk zu dem anderen weiterleiten. TCP-Proxies können mit einer protokollspezifischen Filterung versehen werden und erscheinen "in-situ", mit in Echtzeit geprüften und weitergeleiteten Anwendungsdaten mit nur einer begrenzten Zwischenspeicherung, gegenüber dem Anwendungs-Gateway, welcher einen vollständigen Anwendungs-Kontext sammelt, bevor er die Daten weiterleitet. Die meisten Firewalls verwenden den Begriff Proxy, um ein TCP-Schicht-Relais zu bezeichnen. Ein TCP-Proxy unterstützt separate TCP-Verbindungen zwischen dem Client und dem Firewall und zwischen dem Firewall und dem Server. Um die Transparenz-Anforderungen zu erfüllen, muß ein TCP-Proxy in der Lage sein, einen Verbindungsversuch von dem Client anstelle des Servers anzunehmen.

Aufgabenstellung

[0029] Eine Aufgabe der vorliegenden Erfindung besteht insbesondere darin, eine bessere Leistungsfähigkeit und eine leichtere Konfiguration zu ermöglichen.

[0030] Gemäß einem ersten Aspekt der vorliegenden Erfindung wird vorgeschlagen eine Vorrichtung zum Sicherstellen sicherer Kommunikation zwischen einer anfordernden Entität und einer bedienenden Entität, mit:

– Mitteln zum Annehmen einer Verbindungsanforderung der anfordernden Entität, welche eine Verbindung zu der bedienenden Entität anfordert,
– Mitteln zum Aufbauen einer auf einer Vielzahl von unterschiedlichen Schichten arbeitenden Sitzungs-Verbindung mit der anfordernden Entität,
– Mitteln zum Überprüfen der Kommunikation von der anfordernden Entität hinsichtlich der Übereinstimmung mit einem selektierten Kommunikationsprotokoll auf den oberen der Vielzahl von unterschiedlichen Schichten,
– Mitteln zum Aufbauen einer Sitzungs-Verbindung mit der bedienenden Entität, falls durch die Mittel zum Überprüfen der Kommunikation Übereinstimmung mit dem selektierten Kommunikationsprotokoll festgestellt wurde, und
– Mitteln zum Vermitteln der Kommunikation zwischen der anfordernden Entität und der bedienenden Entität auf der Transportschicht, wenn beide Verbindungen aufgebaut sind,

wobei die Vorrichtung so ausgebildet ist, dass sie für die beiden Entitäten transparent erscheint.

[0031] Gemäß einem zweiten Aspekt der vorliegenden Erfindung wird vorgeschlagen ein Verfahren zum Sicherstellen sicherer Kommunikation zwischen einer anfordernden Entität und einer bedienenden Entität unter Verwendung eines zwischengeschalteten Firewall, mit den Schritten:

– Annehmen einer Verbindungsanforderung der anfordernden Entität, welche eine Verbindung zu der bedienenden Entität anfordert, durch den Firewall,
– Aufbauen einer auf einer Vielzahl von unterschiedlichen Schichten arbeitenden Sitzungs-Verbindung zwischen dem Firewall und der anfordernden Entität,
– Überprüfen der Kommunikation von der anfordernden Entität hinsichtlich der Übereinstimmung

mit einem selektierten Kommunikationsprotokoll auf den oberen der Vielzahl von unterschiedlichen Schichten durch den Firewall,
– falls im voranstehenden Schritt Übereinstimmung mit dem selektierten Kommunikationsprotokoll festgestellt wurde, Aufbauen einer Sitzungs-Verbindung zwischen dem Firewall und der bedienenden Entität und
– wenn beide Verbindungen aufgebaut sind, Vermitteln der Kommunikation zwischen der anfordernden Entität und der bedienenden Entität durch den Firewall auf der Transportschicht,

wobei in sämtlichen Verfahrensschritten der Firewall für die beiden Entitäten transparent erscheint.

[0032] Bevorzugte Ausführungsbeispiele der Erfindung sind in den abhängigen Ansprüchen angegeben.

[0033] Somit wird ein Informationsaustausch zwischen zwei Anwendungs-Entitäten sicher überwacht und gesteuert durch einen Sicherheits-Proxy, welcher in einem Mehrschichten-Kommunikationssystem vorhanden ist. Der Proxy ermittelt Versuche von einem Anforderer, eine Kommunikationssitzung mit einem Server unter Verwendung der unteren Schichten und entsprechend den festgelegten Authentifizierungs-Verfahren aufzubauen. Der Proxy nimmt Anforderer-Verbindungsanforderungen an und baut eine unabhängige Verbindung zu dem Server auf und leitet Kommunikationen in variablen höheren Schichten weiter. Entsprechend einer festgelegten Sicherheitspolitik empfängt in einer Ausführungsform der Proxy transparent Transportpakete und leitet sie weiter. Zusätzlich erfasst der Proxy Sitzungs-Fehlerzustände wie abgefallene Verbindungen und reagiert darauf. Protokolldateneinheiten werden zur Übereinstimmung mit einer Protokoll-Sitzung ermittelt und optional weiterhin dekodiert, um eine zusätzliche anwendungsspezifische Filterung hinzuzufügen.

[0034] In einer Ausführungsform implementiert ein N-Schichten-Kommunikationssystem die 7 Schichten des Open System Interconnect-(OSI)-Protokolls. Der Proxy umfaßt ein Computerprogramm mit einem Verbindungsmanager-Teil und einem Sicherheitsmanager-Teil. Der Proxy wirkt mit Netzwerk-Software zusammen, um einen Kommunikations-Stack zu beeinflussen, um Verbindungsanforderungen an jede Adresse bei bestimmten Anschlüssen zu überwachen. Die Adresse des Anforderers und die Server-Adresse werden anhand einer Zugriffs-Kontrolliste geprüft. Wenn eine der Adressen ungültig ist, schließt der Proxy die Verbindung. Wenn beide gültig sind, wird eine neue Verbindung eingerichtet, so daß der Anforderer und der Server transparent an den Proxy angeschlossen sind. Der Anforderer, der jetzt erwartet, mit dem Server verbunden zu sein, fährt mit der Kommunikation fort und versucht, Verbindungen auf einer höheren Ebene durch Senden einer OSI-Transportanforderung aufzubauen. Der Proxy vermittelt eher den Aufbau der Transportverbindung statt einer aktiven Teilnahme. Die OSI-Transportebenen-Anforderung enthält keine sicherheitsrelevanten Informationen, welche der Proxy abfragen muß und reicht diese Anforderung und die entsprechende Transportantwort daher durch. Der OSI-Proxy faßt jedoch die zwischen dem Client und dem Server während dieser Stufe vereinbarten Verbindungs-Parameter zusammen und zeichnet sie auf.

[0035] Als nächstes tritt der Proxy in die Sitzungs-Aufbauphase ein durch Vermitteln des Aufbaus der Sitzungs-, Präsentations- und Anwendungs-Verbindungen. Der Proxy interpretiert aktiv die Anforderungen für jeden der Dienste der oberen Schichten, wie sie gleichzeitig transportiert und bestätigt werden. Password, unterschiedlicher Name und digitale Signaturen sind sämtlich verfügbar, um die vorhandene Sicherheit zu verbessern.

[0036] Um die Antwort auszuführen, beinhaltet der Proxy Merkmale von Anwendungs-Gateways und Proxies. Um das gewünschte Protokollverhalten zu bewirken, stellt der Proxy seine Verarbeitung derart ein, die Rolle eines Protokoll-Peers in der Schicht, welche für die Antwort geeignet ist, und die Rolle entweder des Auslösers oder des Antwortenden, abhängig davon, auf welcher Seite der Firewall-Vorgang angestoßen wurde, anzunehmen. Der Proxy arbeitet tatsächlich in der Schicht N, wobei N variabel ist, wie durch die Sicherheitspolitik vorgeschrieben, und entsprechend dem erforderlichen Protokollverhalten. Zusätzlich ist die Sicherheitspolitik in dem Sicherheitsmanager im wesentlichen unabhängig von dem Verbindungsmanagementteil des Proxy implementiert.

[0037] Die vorliegende Erfindung bietet eine bessere Leistungsfähigkeit als Anwendungs-Gateways, da eine geringere Verarbeitung erforderlich ist, um Pakete zwischen hohen Ebenen der Protokolle zu übertragen. Sie ist leichter zu konfigurieren und zu verwalten als Anwendungs-Gateways, da es nicht erforderlich ist, das verwendete Kommunikationsprotokoll vollständig zu implementieren. Zusätzlich bietet sie eine höhere Sicherheitsebene als bekannte Proxies und Relais. Ein weiterer Vorteil liegt in der Transparenz des Proxy. Da weder der Anforderer noch der Server den Proxy sieht, sind keine Anforderungen zu modifizieren, um mit dem Proxy zu arbeiten. Ein weiterer Vorteil umfaßt die Fähigkeit, mehrere Anwendungsebenenprotokolle wie X.500-Sitzungen und X.400-Sitzungen mit geringen Änderungen in der Verarbeitung zu verarbeiten oder durchzureichen.

Ausführungsbeispiel

[0038] Es zeigen:

[0039] Fig. 1 ein Blockschaltbild eines die vorliegende Erfindung implementierenden Computersystems;

[0040] Fig. 2 ein kombiniertes Blockschaltbild und logisches Sitzungs-Aufbau-Flußdiagramm einer Ausführungsform der vorliegenden Erfindung;

[0041] Fig. 3a ein logisches Blockdiagramm der Wechselwirkung zwischen Komponenten der Ausführungsform in Fig. 2;

[0042] Fig. 3b ein Blockdiagramm, welches Daten und einen Steuerungsfluß zwischen Komponenten der Ausführungsform in Fig. 2 darstellt; und

[0043] Fig. 4 ein kombiniertes Blockschaltbild und ein Flußdiagramm eines logischen Sitzungs-Aufbaus einer alternativen Ausführungsform der vorliegenden Erfindung.

[0044] In der folgenden detaillierten Beschreibung der bevorzugten Ausführungsformen wird auf die beigefügten Zeichnungen Bezug genommen, welche einen Teil davon bilden, und in welchen bestimmte Ausführungsformen beispielhaft dargestellt sind, in welchen die Erfindung ausführbar ist. Es versteht sich, daß andere Ausführungsformen verwendet werden können und strukturelle Änderungen vorgenommen werden können, ohne vom Umfang der vorliegenden Erfindung abzuweichen.

[0045] Es gab eine Explosion im Wachstum von Computernetzwerken, als Organisationen die Vorteile der Vernetzung ihrer Personalcomputer und Arbeitsplätze erkannten. Zusätzlich fallen diese Netzwerke böswilligen Außenseitern zum Opfer, die in das Netzwerk eindringen, sensitive Informationen lesen und manchmal zerstören. Das solchen Angriffen Ausgesetztsein hat zugenommen, seit Unternehmen an äußere Systeme wie das Internet angeschlossen sind.

[0046] Um sich selbst vor Angriffen durch böswillige Außenseiter zu schützen, gehen Organisationen zu Mechanismen zum Erhöhen der Netzwerksicherheit über. Ein solcher Mechanismus wird beschrieben von Dan Thomsen in "Type Enforcement: the new security model", Proceedings: Multimedia: Full-Service Impact on Business, Education, and the Home, SPIE Ausgabe 2617, Seite 143, August 1996. Thomsen lehrt, daß Modifikationen an dem Kernel eines Betriebssystems vorgenommen werden können, um einen Typ-Durchsetzungs-Schutz hinzuzufügen.

[0047] Die Typ-Durchsetzung fügt eine zusätzliche Schutzebene in den Vorgang des Dateizugriffs ein. Der Grundgedanke der Typ-Durchsetzung ist, jedem Programm nur die Erlaubnisse zu geben, die das Programm benötigt, um seine Aufgabe zu erfüllen. Dieses Konzept wird als "kleinste Privilegien" bezeichnet. Typ-Durchsetzung arbeitet durch Gruppieren von Vorgängen in Klassen, basierend auf dem kleinsten Privileg. Jede Gruppierung wird als ein Bereich bezeichnet. Sämtliche Dateien in dem System sind ebenfalls in Klassen zusammengefaßt, welche als Typen bezeichnet werden, und eine Tabelle (die Bereichs-Definitionstabelle oder DDT) wird geschaffen, welche festlegt, wie ein Vorgang jede Art von Datei ansprechen kann.

[0048] Ein Typ-Durchsetzungs-Ansatz für die Sicherheit ist durchaus hilfreich für das BSD 4.4 UNIX-Betriebssystem, da es bei der ungesicherten Version des Systems, sobald ein Vorgang Privilegien erhält, diese Privilegien nutzen kann, um auf andere Netzwerk-Dateien zuzugreifen. Dies kann zu einem gefährlichen Bruch der Netzwerksicherheit führen.

[0049] Zusätzlich lehrt Thomsen, daß die Typ-Durchsetzung verwendbar ist, um abgesicherte Pipelines zwischen Programmen einzurichten. Die Typ-Durchsetzung erzeugt solch eine abgesicherte Pipeline durch Steuern des Zugriffs zwischen Programmen. D.h., jedes Programm hat nur die Erlaubnis, aus der dem Programm vorausgehenden Stufe zu lesen und in der dem Programm folgenden Stufe zu schreiben. Keine Stufe der Pipeline kann umgangen werden. Thomsen merkt an, daß um abgesicherte Pipelines aufgebaute Anwendungen leichter zu analysieren und ihr sicherer Betrieb zu bestätigen ist.

[0050] Schließlich gibt Thomsen eine Firewall-Anwendung der Typ-Durchsetzung an. Thomsen lehrt, daß ein sicherer Computer verwendet werden kann, um ein privates Netzwerk mit mehreren Arbeitsplätzen an ein öffentliches Netzwerk anzuschließen. Ein auf dem sicheren Computer ablaufendes Protokollpaket wie TCP/IP implementiert ein Kommunikationsprotokoll, das zum Kommunizieren zwischen jedem Arbeitsplatz und dem sicheren Computer verwendet wird. Thomsen lehrt, daß ein doppelter Protokoll-Stack genutzt werden kann, um die Bewegung von Informationen durch den Firewall zu begrenzen. Auf dem sicheren Computer ablaufender Programmcode wird zur Kommunikation durch das private Netzwerk zu dem Arbeitsplatz-Protokollpaket verwendet. Solch ein System wird gegenwärtig unter dem Namen Sidewinder von der Anmelderin verkauft.

[0051] In einer Ausführungsform ist der sichere Computer eine Intel-Pentium-basierte Maschine, die mit einer stabilen Form von BSD386 Unix läuft. Ein System, basierend auf einem 90MHz Pentium-Mikroprozessor mit 32 Megabyte Speicher, 2 Gigabyte Festplatten-Platz, einem DAT-Band zur Sicherung und CD-ROM zum Laden von Software wurde als adäquat festgestellt. Ebenso wird auf dem sicheren Computer ablaufender Programmcode durch eine Schnittstelle zu einem öffentlichen Netzwerk zur

Kommunikation mit einem öffentlichen Netzwerk wie dem Internet verwendet. In einer Internet-Ausführungsform ist der zum Kommunizieren mit dem Internet verwendete Programmcode Teil eines Satzes von Internet-Protokollen, welche mit Computern im Internet durch eine Internet-Verbindung kommunizieren. In einer Ausführungsform können beim Kommunizieren mit unterschiedlichen Entitäten im Internet unterschiedliche Protokolle verwendet werden. In einer Ausführungsform wird ein in den Internet-Protokollen arbeitendes äußeres Mantel-Paket (Top Wrapper Package) verwendet, um auf dem externen, öffentlichen Netzwerk aufzusitzen, so daß Informationen über externe Probleme aufgezeichnet werden können.

[0052] Die vorliegende Erfindung ist eine Erweiterung des Sidewinder-Produktes. Wie in Fig. 1 allgemein bei **110** gezeigt, umfaßt ein Computersystem einen Prozessor **112**, der an einen wahlfreien Zugriffsspeicher RAM **114** gekoppelt ist. Während nur ein einzelner Bus **116** gezeigt ist, welcher den RAM **114** und den Prozessor **112** mit einem Kommunikationsanschluß **118** und einem Disk-Laufwerk oder einem anderen Speichermedium **120** verbindet, ist für den Durchschnittsfachmann erkennbar, daß er mehrere unterschiedliche Busse in einer Standard-Personalcomputerarchitektur repräsentiert. Der Kommunikationsanschluß repräsentiert verschiedene Kommunikationsoptionen in Computersystemen, wie Ethernet-Karten, Modems und andere Kommunikations-Geräte.

[0053] In Fig. 2 ist eine Computerprogrammerweiterung für das Sidewinder-Produkt allgemein mit **210** bezeichnet. Das Computerprogramm ist allgemein auf dem Disk-Laufwerk **120** gespeichert und läuft ab oder wird ausgeführt durch den Prozessor **112** aus dem RAM **114**. Es ist anzumerken, daß das Disk-Laufwerk **120** hier verwendet wird, um unterschiedliche Speichermedien darzustellen, durch welche das Computerprogramm **210** gespeichert und verteilt werden kann. Es stellt ebenfalls ein Kommunikationsmedium dar, auf welchem das Programm vorübergehend gespeichert werden kann, während es zu dem Computersystem **110** übertragen wird. Das Computerprogramm **210** umfaßt weiterhin einen Proxy **212**, welcher verwendet wird, um Kommunikationen zu verarbeiten, welche mit unterschiedlichen Arten von OSI-Anwendungsprotokollen, wie dem gezeigten X.500-Protokoll, übereinstimmen. Weiterhin sind in Fig. 2 ein Client **214** und ein Server **216** gezeigt, für welche Verbindungen und Datenübertragungen weiter unten beschrieben werden.

[0054] Das Sidewinder-Sicherheitssystem weist besondere TCP/IP-Netzwerk-Modifikationen auf, welche es erlauben, eine TCP-Verbindungsanforderung anzunehmen, auch wenn die Daten nicht an das System adressiert sind. Das Sidewinder kann die Daten verifizieren und eine weitere, unabhängige Sitzung mit dem realen Zielgerät unter Verwendung der innerhalb der Originalanforderung des Senders festgelegten Zieladresse aufbauen. Dies ist das zum Implementieren der transparenten Funktionalität für die meisten Sidewinder-Anwendungen genutzte Verfahren.

[0055] Die Verwendung eines solchen TCP-Proxy ist ausreichend, um zwei End-Stationen zu erlauben, unter Verwendung von OSI-Anwendungen über TCP/IP zu kommunizieren. Es ist jedoch nicht sehr sicher. OSI-basierte Anwendungen arbeiten unter Verwendung von vier 14) weiteren Schichten von OSI-Protokollen, wie in Fig. 3a erkennbar ist: Transport-, Sitzungs-, Präsentations- und Anwendungs-Diensten; und dann schließlich der Anwendungsdaten (wie FTAM, X.400, X.500, etc.). Wie in dieser Darstellung des Computerprogramms erkennbar ist, arbeitet der Proxy in den oberen Schichten von OSI, während das Relais Transportdaten in der Transportschicht überträgt.

[0056] In Fig. 3b sind Darstellungen von Modulen und Komponenten des Proxy gezeigt. Ein Client überträgt Transportdaten oder PDUs zu einem TCP-Stack in dem Programm. Der Stack leitet die Daten zu dem Relais weiter, welches sie wiederum zu einem Verbindungs-Manager weiterleitet. Der Verbindungsmanager arbeitet mit einer Sicherheitsüberwachung, welche die Daten auf Übereinstimmung mit vorbestimmten Bedingungen, die oben und weiter unten beschrieben sind, überwacht. Sie liefert dann Steuerungsinformationen zu dem Verbindungs-Manager, welcher wiederum das Relais steuert und es anweist, ob es Verbindungen durch einen weiten Stack zu einem Server aufbauen soll.

[0057] Aufgrund der allgemein öffentlich angenommenen TPO-OSI-Transportdienste über TCP/IP kann ein TSAP verwendet werden. TSAP ist der Begriff, der benutzt wird, um die Zuordnung zwischen einem Sitzungs-Anbieter und einem bestimmten Transport-Endpunkt (vergleichbar mit dem Konzept der Anschlüsse bei TCP) zu beschreiben. Dies ist ebenfalls zutreffend für die OSI-Sitzungs-(SSAPs genannt) und die OSI-Präsentations-(PSAPs genannt)-Dienste. Wie bei TCP sind TSAP-, SSAP- und PSAP-Definitionen für besondere Anwendungen nicht direkt einer bestimmten Anwendung zugewiesen. Diese bleiben übrig für die einzelnen Anwendungs-Administratoren.

[0058] Ebenso wie bei TCP verwenden OSI-TPO-Sitzungs- und Präsentations-Protokolle sämtlich ein formales Verbindungs/Unterbrechungs-Protokoll. Jedoch nur in dem Fall von OSI-TPO müssen die Verbindungen aufgebaut sein, bevor Daten gesendet werden können. OSI-Sitzungs- und OSI-Präsentations-Schichten erlauben den Benutzern, Daten in die aktuelle Verbindungsan-

forderung einzuschließen. Dies beseitigt die Notwendigkeit zum Senden einer Verbindungsanforderung, Warten auf Antwort ... Senden einer Verbindungsanforderung, Warten auf Antwort ... in diesen zwei Schichten.

[0059] Eine einfache Lösung scheint das Aufbauen eines Proxy, welcher verifiziert, daß die TCP-Daten OSI-Transport- und/oder OSI-Sitzungs-Protokolldaten enthalten. Hierin liegt jedoch das Sicherheitsproblem. Dies läßt die Tür offen für eine End-Station, um durch den Proxy unter Verwendung jeder Art von OSI-Anwendungsprotokoll (d.h., FTAM, X.400, X.500, etc.) zu kommunizieren. Um dieses auszugleichen muß der Proxy den Präsentations-Kontext innerhalb der Präsentations-Verbindungsanforderung prüfen. Der Präsentations-Kontext identifiziert die erwartungsgemäß zu nutzenden OSI-Anwendungs-Entitäten während dieser Sitzung. Der Proxy kann dann ausreichend sicher sein, daß diese Sitzung mit der erwarteten Anwendung kommuniziert.

[0060] Wenn der Proxy jedoch die Präsentations-Kontexte verifizieren soll, dann müssen mehr Daten geprüft werden und ein Informationsaustausch muß zwischen dem Anforderer und dem Proxy stattfinden. Genauso, wie es bei TCP stattfindet, muß der Proxy auf die OSI-Transport-Verbindungsanforderung antworten, auch wenn sie nicht an ihn adressiert ist. Dies soll das anfragende Gerät veranlassen, den Sitzungs- und Präsentations-Verbindungsanforderungsrahmen zu senden, welcher eine Identifizierung der Anwendung beinhaltet. Der Proxy kann jetzt die Präsentations-Kontexte verifizieren und eine Sitzung mit dem Zielgerät initiieren.

[0061] Um die Proxy-Verarbeitung zu vervollständigen und sicherzustellen, daß nur bestimmte OSI-Anwendungsdaten zu den Sitzungen weitergeleitet werden, prüft die Proxy-Software schließlich kontinuierlich die OSI-Anwendungsebenen-Protokolle innerhalb der Datenrahmen. Z.B. verifiziert in einer Ausführungsform ein X.500-Proxy, daß während der Sitzung ausgetauschte Daten mit einem bestimmten X.500-Protokoll übereinstimmen. Der Proxy erfüllt jetzt die oben beschriebenen Anforderungen. Der Proxy antwortet auf Client- und Server-Anforderungen an andere Systeme, evaluiert die Anforderung und baut separate Verbindungen mit dem Zielsystem auf. Dies alles geschieht transparent für die End-Stationen, auf welchen Anwendungs-Entitäten ablaufen, da sich in den Geräten nichts ändert. Alle Adressen (IP, OSI TSAP, OSI SSAP etc.) und Software bleiben gleich.

[0062] Diese Lösung ist ebenfalls sehr vielseitig, da der Proxy modifiziert ist, um die meisten OSI-Anwendungsebenenprotokolle "durchzureichen". Er verarbeitet z.B. X.400-Sitzungen (wie ein P7-Client zur Mitteilungs-Speicherung) in gleicher Weise mit geringfügigen Änderungen in der Verarbeitung. Zusätzlich erlaubt er die Unterstützung von X.500 DISP-Sitzungen durch das System. Dies ist mit der vollständigen DSA-Anwendungs-Gateway-Lösung nicht möglich.

[0063] Der X.500-Proxy erlaubt allen X.500-definierten Protokollen, einschließlich dem Directory Service Protocol (DSP), Directory Access Protocol (DAP), Directory Information System Protocol (DISP) und dem Directory Operational Binding Management Protocol (DOP), bei den X.500-Geräten den Sidewinder zu durchlaufen. Der X.500-Proxy stellt eine oder mehrere der folgenden Funktionen bereit, welche in den in Fig. 3b gezeigten Komponenten verteilt sind:

- Ermitteln aller hereinkommenden Anforderungen für neue X.500-Sitzungen, um die Übereinstimmung mit konfigurierten Authentifizierungs-Verfahren sicherzustellen.
- Authentifizieren der Signatur und X.500-Zertifikate, die in dem strengen Authentifizierungs-BIND-Vorgang zum Sitzungsaufbau zwischen Geräten vorgesehen sind.
- Ermitteln und Filtern von Protokoll-Dateneinheiten (PDUs) zur Übereinstimmung mit OSI-Protokollen.
- Erfassen von Sitzungs-Fehlerbedingungen wie abgefallenen Verbindungen.
- Bereitstellen von Filterfunktionen für die X.500-Vorgangs-Attribut- und Wertebasierten Sitzungs-PDUs.

[0064] Sobald eine Verbindung zwischen zwei Geräten in unterschiedlichen Netzwerken aufgebaut ist, leitet der Proxy alle X.500-PDUs (Anforderungen und ihre Antworten) transparent zu den zwei Geräten weiter. Daten-Elemente werden zum Prüfen der Übereinstimmung mit der initiierten Protokoll-Sitzung dekodiert, es wird jedoch keine weitere Ermittlung während der Sitzung ausgeführt. Eine zusätzliche Filterung kann bei Bedarf ausgeführt werden.

[0065] Bei der Verwendung dieses Proxy-Aufbaus müssen weder der Client noch der Server irgendwelche Konfigurationsinformationen oder Software ändern, um den Firewall zu implementieren. Das ist der Grund, warum diese Lösung als "transparent" bezeichnet wird. In einer alternativen Ausführungsform maskieren (verstecken) der Anforderer, der Server oder beide Systeme ihre Adressen voreinander durch Ansprechen des Sidewinder als Server.

[0066] Es gibt zwei Hauptkomponenten der vorliegenden Erfindung. Die Proxy-Kommunikationskomponente, welche die Verbindungen zwischen den zwei Geräten in jedem Netzwerk handhabt, und die Filterkomponente. Es gibt drei beschriebene Proxy-Kommunikationskomponenten-Ausführungsformen:

- 1: OSI-Transportschicht-Proxy mit "transparenter Bridge"

– 2: OSI-Transportschicht-Proxy mit "transparenter Bridge" mit vorbestimmten Transportschicht-Antworten (positive und negative)
– 3: OSI-Sitzungsschicht-Proxy mit "transparenter Bridge" (dies bedeutet vollständige Transportschicht-Dienste)

[0067] Der Proxy mit transparenter Bridge arbeitet direkt über den TCP/IP-Protokoll-Stack, wie in Fig. 2 gezeigt. Die Proxy-Software hat eine Schnittstelle mit TCP/IP unter Verwendung der Sockel-Bibliothek und weist den TCP/IP-Stack an, auf TCP-Verbindungsanforderungen an jede IP-Adresse an bestimmten Anschlüssen (wie 102, 17003, etc.) zu warten. Kommunikationsschritte zwischen dem Client **214**, dem Proxy **212** und dem Server **216** sind zwischen jedem Block gezeigt und in einem Kreis nahe einer Beschreibung jedes Schrittes entsprechend der Reihenfolge, in welcher sie durch das Programm **210** gesteuert werden, numeriert.

[0068] Die Sidewinder-TCP/IP-Software antwortet auf die Verbindungsanforderung und leitet die rufende Information zu dem Proxy weiter. Der Proxy verwendet diese Information zum Prüfen der Anforderer-IP-Adresse (Quelle) und der Server-IP-Adresse (Ziel) anhand einer Zugriffs-Steuerungsliste (ACL). Wenn eine Adresse ungültig ist, schließt der Proxy die Verbindung. Wenn beide Adressen gültig sind, versucht der Proxy, eine neue Verbindung mit dem Server in dem Ziel-Netzwerk aufzubauen. Der Anforderer bemerkt nichts von dieser unabhängig stattfindenden Verbindung. Soweit er betroffen ist, hat er bereits die Kommunikation mit dem Ziel-Server begonnen. Der Anforderer sollte in dem Vorgang sein, eine OSI-Transport(T_CONNECT)-Anforderung zu senden. Der Proxy verifiziert diese Anforderung, um sicherzustellen, daß es ein T_CONNECT ist, und leitet sie einfach zu dem Server weiter. Der Server sendet dann eine T_CONNECT-Antwort, welche verifiziert und direkt zurück zu dem Anforderer weitergeleitet wird. Als nächstes soll der Anforderer einen Rahmen mit der Sitzungs-Verbindungsanforderung, der Präsentations-Verbindungsanforderung (welche die Anwendungs-Kontexte beinhaltet) und der X.500-Protokoll-BIND-Anforderung senden.

[0069] Diese X.500-BIND-Anforderung enthält die Authentifizierungsinformation, die der X.500-Server benötigt, um zu identifizieren, daß dieser Benutzer auf die Datenbank dieses Directory-Servers zugreifen darf. Die X.500-BIND kann drei (3) unterschiedliche Formen der Authentifizierung enthalten: keine, einfach und streng. Keine bedeutet, daß der Benutzer keinerlei Authentifizierung liefert. Einfach bedeutet, daß der Benutzer sich selbst identifiziert hat und ein Klartext-Passwort für den Server liefert, um sich selbst zu verifizieren. Streng bedeutet, daß eine Form eines verschlüsselten Passwortes, der X.500-Servername oder Authentifizierungsinformationen und meistens eine Form einer digitalen Signatur vorhanden sind. Der Proxy übergibt diese BIND-PDU zu der Filterkomponente zum Verifizieren. Die Proxy-Filterkomponente handhabt alle drei BIND-Möglichkeiten. Das Filter unterstützt eine Konfigurationsdatei, welche enthält, welche Art von Authentifizierung zugelassen ist, wer zugelassen ist und welche mögliche Zurückweisung bei einem Fehler zurückgegeben wird. Die Filterkomponente verarbeitet dann das BIND und gibt den Status zu der Kommunikationskomponente zurück. Basierend auf dem Status kann der Proxy das BIND zu dem X.500-Server weiterleiten oder kann beide Sitzungen abbrechen und die Verbindungen schließen. Der beim Fehler zurückzugebende Status ist konfigurierbar.

[0070] Da der Proxy OSI-Transport-, Sitzungs-, Präsentations-, ACSE- oder ROSE-Schichten nicht implementiert, muß er manuell geeignete Antworten bilden, um eine vorhandene Verbindung zurückzuweisen oder auch möglicherweise abzubrechen. Wenn der Proxy zum Beispiel eine aufgebaute TCP-Verbindung und eine TPO-Verbindung aufweist und dann eine Präsentations-P_CONNECT-Anforderung mit einer ACSE A-ASSOCIATE-Anforderung für einen X.400-Präsentations-Kontext empfängt, muß der Proxy eine Zurückweisung für diese Anforderung erzeugen und die Verbindungen schließen. Der Proxy kann grob sein und einfach die IP-Verbindung schließen, aber dies ist nicht wirklich geeignet. Der Anforderer kann ein Netzwerkproblem annehmen und die Verbindung erneut versuchen. Die geeignet Antwort ist, eine ACSE A-ASSOCIATE-Antwort "zurückgewiesen (dauerhaft)" (rejected (permanently)) aufzubauen. Jetzt werden, da der Proxy eine Antwort erzeugen muß (nicht nur eine von der anderen Seite durchreichen muß), alle geeigneten OSI-Ebenen-Antworten in dieser PDU zusammengefaßt. Dies beinhaltet die Präsentations-P_CONNECT-Antwort mit "Benutzer-Zurückweisungs-Status, den Sitzungs-Ebenen-S-CONNECT-Bestätigungs(Zurückweisungs)-Status" Zurückweisung durch SS-Provider und die Transport-T-DATA-Bestätigung. Dann werden die RFC1006-Antworten erzeugt und schließlich zum Übertragen zum TCP abgegeben. Geeignete Time-Out-Werte werden in dem Fall, daß die Verbindungen verlorengehen, eingestellt, um aufzuräumen. All dies wird ausgeführt, nur um die "korrekte" Antwort zu erzeugen.

[0071] Die zweite Ausführungsform modifiziert die Wirkungsweise der ersten Ausführungsform des Proxy durch Bereitstellen der Möglichkeit für das Sidewinder-System, direkt auf die TCP/IP-Sitzungs- und OSI-Transportverbindungs-Anforderungen des Anforderers zu antworten, um X.500-Kommunikationen zu empfangen, wie in Fig. 4 gezeigt. Der Server liefert als nächstes die T-CONNECT-Anforderung zurück, bei welcher der Proxy die BIND (T_DATA und alles), wie von dem Anforderer empfangen, sendet.

Der Server soll dann die Anforderung verarbeiten und eine geeignete Antwort zurück zu dem Sidewinder-System liefern. An diesem Punkt muß das Sidewinder-System die Antwort prüfen, um zu bestimmen, ob der Server die BIND-Anforderung annimmt oder zurückweist, und die Antwort für den Client entsprechend handhaben.

[0072] Da diese Ausführungsform nicht einen vollständigen Transportschicht-Dienst implementiert, werden einfache generische Antworten auf der Basis dessen, was der Standard-Transportschicht-Dienst bereitstellt, verwendet. Zum Beispiel implementieren gegenwärtige ISODE-Konsortium-Produkte Max-TSDU-Größen von 2041 Bytes. Die fraglichen Parameter sind bei der Systeminstallation mit voreingestellten Werten konfigurierbar und im Sidewinder-Administrationsmodus modifizierbar. Das Problem ist jedoch, daß, sobald der Sidewinder-Proxy eine Größe mit dem Anforderer vereinbart hat, die Transport-Parameter NICHT in der Sidewinder-Zu-Server-Transportverbindung neu vereinbart werden dürfen. Wenn der Server einen der Transport-Parameter herabsetzt, muß diese Verbindung abgebrochen und TCP-Verbindung geschlossen werden, da eine kompatible Verbindung nicht aufgebaut werden kann.

[0073] Die dritte Ausführungsform führt eine "modifizierte" vollständige Transportschicht-Dienstkomponente zum Betrieb zwischen der TCP/IP-Netzwerkschicht und der Proxy-Software ein. Diese Phase erfordert Änderungen bei den Proxy-Komponenten der vorherigen Phase. Die OSI-Transportdienst-Schicht wird portiert und der Proxy zur Verwendung der zum Kommunizieren mit dem OSI-Transportschicht-Dienst verfügbaren API (gegenüber der Sockel-API für TCP/IP) modifiziert.

Patentansprüche

1. Vorrichtung zum Sicherstellen sicherer Kommunikation zwischen einer anfordernden Entität und einer bedienenden Entität, mit:
– Mitteln zum Annehmen einer Verbindungsanforderung der anfordernden Entität, welche eine Verbindung zu der bedienenden Entität anfordert,
– Mitteln zum Aufbauen einer auf einer Vielzahl von unterschiedlichen Schichten arbeitenden Sitzungs-Verbindung mit der anfordernden Entität,
– Mitteln zum Überprüfen der Kommunikation von der anfordernden Entität hinsichtlich der Übereinstimmung mit einem selektierten Kommunikationsprotokoll auf den oberen der Vielzahl von unterschiedlichen Schichten,
– Mitteln zum Aufbauen einer Sitzungs-Verbindung mit der bedienenden Entität, falls durch die Mittel zum Überprüfen der Kommunikation Übereinstimmung mit dem selektierten Kommunikationsprotokoll festgestellt wurde, und
– Mitteln zum Vermitteln der Kommunikation zwischen der anfordernden Entität und der bedienenden Entität auf der Transportschicht, wenn beide Verbindungen aufgebaut sind,
wobei die Vorrichtung so ausgebildet ist, dass sie für die beiden Entitäten transparent erscheint.

2. Vorrichtung nach Anspruch 1, bei welcher die Mittel zum Überprüfen der Kommunikation so ausgebildet sind, dass sie Protokoll- und Sicherheitsverletzungen auf einer ersten Schicht identifizieren und auf der Vielzahl von unterschiedlichen Schichten antworten.

3. Vorrichtung nach Anspruch 2, bei welcher die Mittel zum Annehmen einer Verbindungsanforderung so ausgebildet sind, dass sie eine anfordernde Entität durch Erzeugen einer Antwort auf der Präsentationsschicht, der Sitzungsschicht und der Transportschicht, welche Schichten die Vielzahl von unterschiedlichen Schichten aufweist, zurückweist.

4. Vorrichtung nach Anspruch 2 oder 3, bei welcher eine Open-Systems-Interconnect(OSI)-Kommunikation implementiert ist.

5. Vorrichtung nach Anspruch 4, bei welcher die Mittel zum Überprüfen der Kommunikation so ausgebildet sind, dass sie Protokollinformationen und Daten (PDUs) in den OSI-Transport-, Sitzungs-, Präsentations- und Anwendungsschichten überwachen, und die Mittel zum Vermitteln der Kommunikation so ausgebildet sind, dass sie derartige Protokollinformationen und Daten zu der bedienenden Entität weiterleiten.

6. Vorrichtung nach mindestens einem der vorangegangenen Ansprüche, mit Mitteln zum Überwachen der Verbindungszeit und zum Erfassen von Sitzungsfehlerzuständen.

7. Vorrichtung nach mindestens einem der vorangegangenen Ansprüche, welche als ein RFC-1006-Proxy für Open-Systems-Interconnect(OSI)-Anwendungen vorgesehen ist, die Association-Control-Service-Elemente (ACSE) und Remote-Operations-Service-Elemente (ROSE) verwendet.

8. Vorrichtung nach mindestens einem der vorangegangenen Ansprüche, bei welcher die Mittel zum Überprüfen der Kommunikation so ausgebildet sind, dass sie von der anfordernden Entität kommunizierte Daten in Übereinstimmung mit unterstützen Protokoll-Standards und Beibehaltung einer festgelegten Sicherheitspolitik überwachen und selektiv modifizieren, wobei die Mittel zum Aufbauen einer Sitzungs-Verbindung unter der Steuerung der Mittel zum Überprüfen der Kommunikation arbeiten.

9. Verfahren zum Sicherstellen sicherer Kommu-

nikation zwischen einer anfordernden Entität und einer bedienenden Entität unter Verwendung eines zwischengeschalteten Firewall, mit den Schritten:
– Annehmen einer Verbindungsanforderung der anfordernden Entität, welche eine Verbindung zu der bedienenden Entität anfordert, durch den Firewall,
– Aufbauen einer auf einer Vielzahl von unterschiedlichen Schichten arbeitenden Sitzungs-Verbindung zwischen dem Firewall und der anfordernden Entität,
– Überprüfen der Kommunikation von der anfordernden Entität hinsichtlich der Übereinstimmung mit einem selektierten Kommunikationsprotokoll auf den oberen der Vielzahl von unterschiedlichen Schichten durch den Firewall,
– falls im voranstehenden Schritt Übereinstimmung mit dem selektierten Kommunikationsprotokoll festgestellt wurde, Aufbauen einer Sitzungs-Verbindung zwischen dem Firewall und der bedienenden Entität und
– wenn beide Verbindungen aufgebaut sind, Vermitteln der Kommunikation zwischen der anfordernden Entität und der bedienenden Entität durch den Firewall auf der Transportschicht, wobei in sämtlichen Verfahrensschritten der Firewall für die beiden Entitäten transparent erscheint.

10. Verfahren nach Anspruch 9, mit den weiteren Schritten:
– Prüfen einer Adresse der anfordernden Entität anhand einer Liste autorisierter Adressen durch den Firewall und
– Modifizieren der Antwort an die anfordernde Entität in Abhängigkeit von dem Ergebnis der Prüfung durch den Firewall.

11. Verfahren nach Anspruch 10, bei welchem der Prüfungsschritt, der Modifizierungsschritt und der Schritt zur Annahme einer Verbindungsanforderung auf einer der oberen der Vielzahl von unterschiedlichen Schichten ausgeführt werden.

12. Verfahren nach mindestens einem der Ansprüche 9 bis 11, bei welchem der Schritt, die Kommunikation von der anfordernden Entität zu überprüfen, eine Überprüfung hinsichtlich der Übereinstimmung mit einem Open-Systems-Interconnect(OSI)-Kommunikationsprotokoll umfasst.

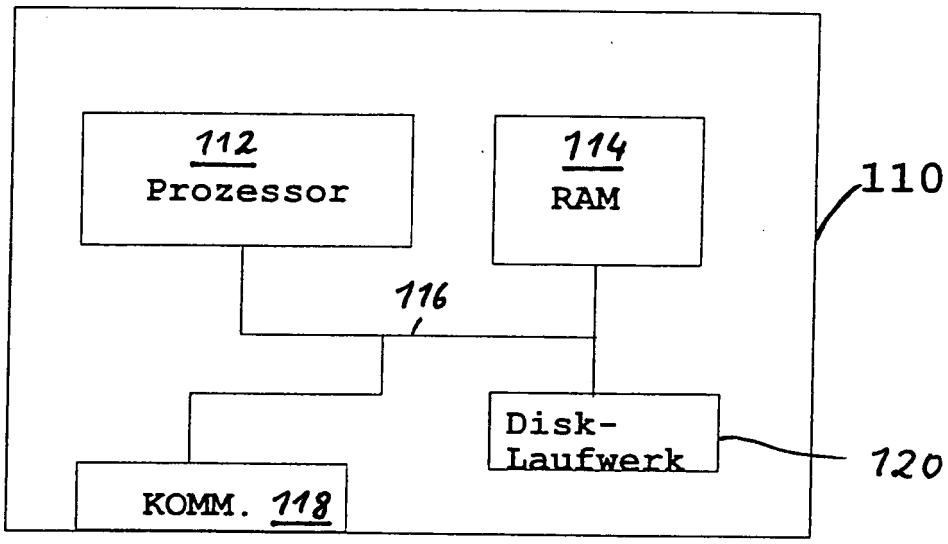
13. Verfahren nach Anspruch 12, bei welchem der Schritt, die Kommunikation von der anfordernden Entität zu überprüfen, eine Überprüfung auf einer OSI-Transportschicht umfasst.

14. Verfahren nach mindestens einem der Ansprüche 9 bis 13, ferner mit den Schritten, die Verbindungszeit zu überwachen und auf Sitzungsfehlerbedingungen zu antworten.

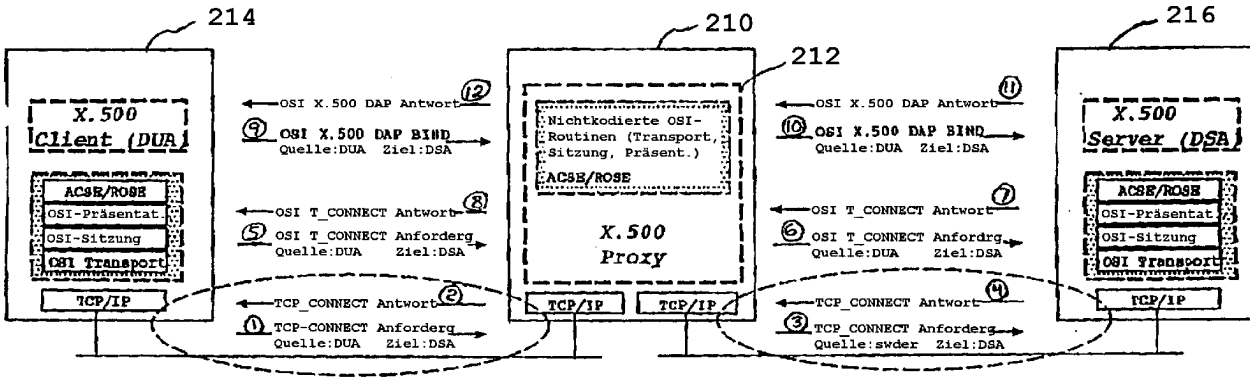
15. Speichermedium mit einem darauf gespeicherten Computerprogramm zur Durchführung eines Verfahrens nach mindestens einem der Ansprüche 9 bis 14.

Es folgen 4 Blatt Zeichnungen

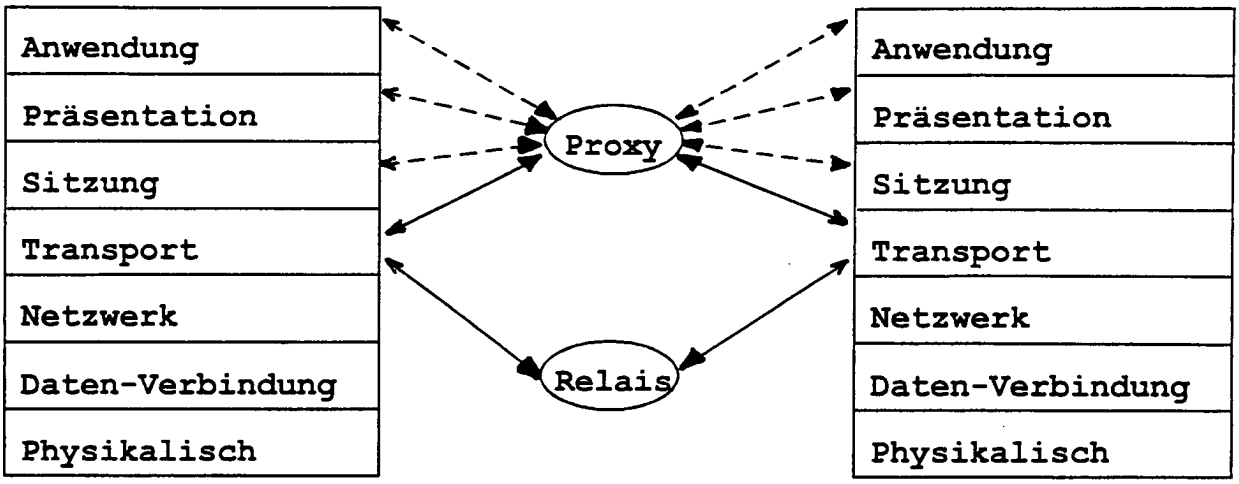
Anhängende Zeichnungen



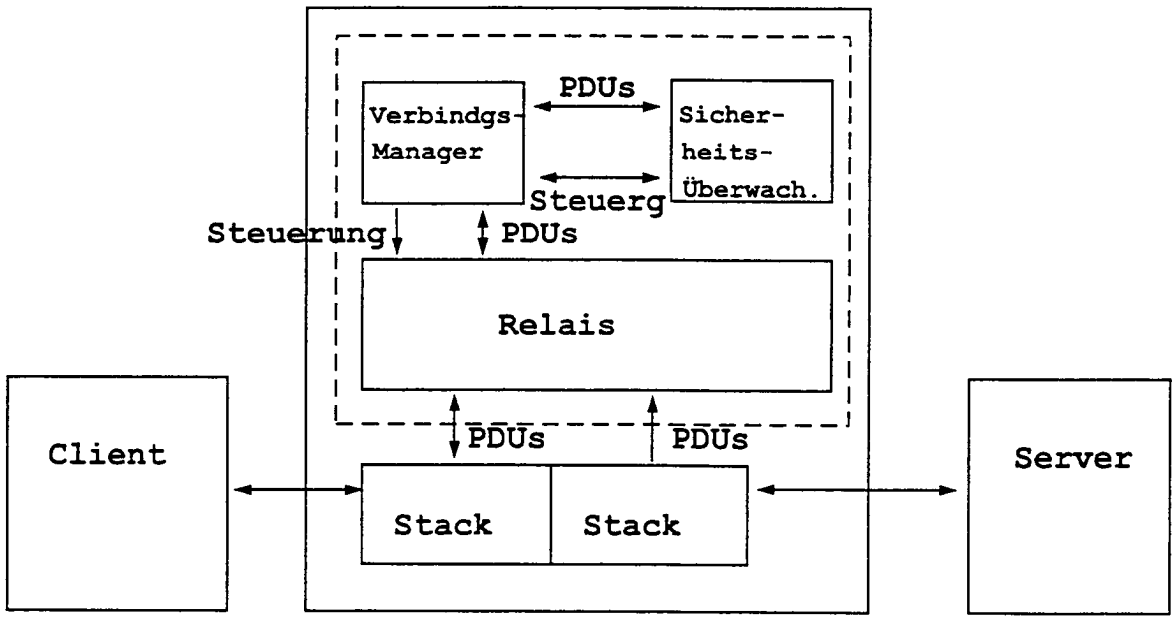
Figur 1


214
210
212
216
X.500
Client (DUA)
ACSE/ROSE
OSI-Präsentat.
OSI-Sitzung
OSI Transport
TCP/IP
OSI X.500 DAP Antwort
OSI X.500 DAP BIND
Quelle:DUA Ziel:DSA
OSI T_CONNECT Antwort
OSI T_CONNECT Anforderg
Quelle:DUA Ziel:DSA
TCP_CONNECT Antwort
TCP-CONNECT Anforderg
Quelle:DUA Ziel:DSA
Nichtkodierte OSI-Routinen (Transport, Sitzung, Präsent.)
ACSE/ROSE
X.500
Proxy
TCP/IP
TCP/IP
OSI X.500 DAP Antwort
OSI X.500 DAP BIND
Quelle:DUA Ziel:DSA
OSI T_CONNECT Antwort
OSI T_CONNECT Anfordrg
Quelle:DUA Ziel:DSA
TCP_CONNECT Antwort
TCP_CONNECT Anforderg
Quelle:swder Ziel:DSA
X.500
Server (DSA)
ACSE/ROSE
OSI-Präsentat.
OSI-Sitzung
OSI Transport
TCP/IP

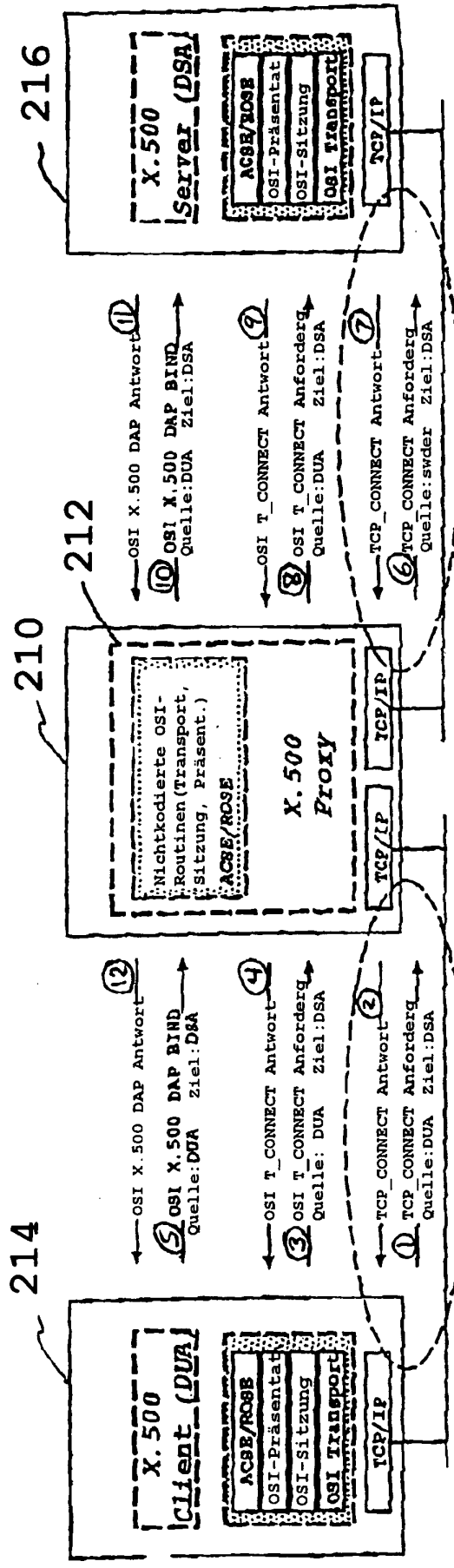
Figur 2



Figur 3a



Figur 3b


216
X.500
Server (DSA)
ACSE/ROSE
OSI-Präsentat
OSI-Sitzung
OSI Transport
TCP/IP
212
210
X.500
Proxy
Nichtkodierte OSI-Routinen (Transport, Sitzung, Präsent.)
ACSE/ROSE
TCP/IP
214
X.500
Client (DUA)
ACSE/ROSE
OSI-Präsentat
OSI-Sitzung
OSI Transport
TCP/IP
OSI X.500 DAP Antwort
OSI X.500 DAP BIND
Quelle:DUA Ziel:DSA
OSI T_CONNECT Antwort
OSI T_CONNECT Anforderg
Quelle: DUA Ziel:DSA
TCP_CONNECT Antwort
TCP_CONNECT Anforderg
Quelle:DUA Ziel:DSA
Quelle:swder Ziel:DSA

Figur 4