

Unterrichtsmitschrift

Betreuung von IT-Systemen

Michael Puff

2009-05-24

Oskar-von-Miller Schule Kassel
Fachinformatiker für Anwendungsentwicklung

Vorbemerkung

Zum Inhalt

Dieses Dokument folgt dem Unterrichtsinhalt von Herrn Sobiroj im Fach *Betreuung von IT-Systemen*. Die eigenen Unterrichtsmitschriften sind durch Texte und Grafiken aus den angegebenen Quellen ergänzt worden.

Diese Ausarbeitung erhebt keinen Anspruch auf Vollständigkeit.

Kontaktmöglichkeiten

Homepage: <http://www.michael-puff.de>

E-Mail: mail@michael-puff.de

Copyright Hinweis

DIESES DOKUMENT STEHT UNTER DER CREATIVE COMMON LICENCE. DAS DOKUMENT DARF ZU DEN FOLGENDEN BEDINGUNGEN WEITER VERVIELFÄLTIGT UND VERBREITET WERDEN. DER NAME DES AUTORS/RECHTEINHABERS (MICHAEL PUFF) IST ZU NENNEN. DIESES DOKUMENT DARF NICHT BEARBEITET ODER IN ANDERER WEISE VERÄNDERT WERDEN.

Inhaltsverzeichnis

1	Firewalls	7
1.1	Funktion und Aufgaben von Firewall-Systemen	7
1.2	Sicherheitspolitik	8
1.2.1	Sicherheitsanforderungen	8
1.2.2	Kommunikationsanforderungen	9
1.2.3	Maßnahmen bezüglich Organisation, Personal und Infrastruktur	9
1.3	Typen von Firewalls	10
1.3.1	Paketfilter	10
1.3.2	Application Gateway (Proxy)	10
1.3.3	Gegenüberstellung Paketfilter – Application Gateway	12
1.4	Komponenten einer Firewall	12
1.5	Technische Implementierung	13
1.5.1	Schichtenmodell	13
1.5.2	Der Paketfilter	13
1.5.3	Application Gateway	14
1.5.4	Vergleich	14
1.6	Typische Netzwerk Topologien	15
1.6.1	Topologie nur mit Paketfilter	15
1.6.2	Topologie nur mit Application Gateway	17
1.6.3	Screened Subnet Topologie mit Single Homed Application Gateway	18
1.6.4	Screened Subnet Topologie mit Multi Homed Application Gateway	19
1.6.5	Beliebiges Konzept	20
2	Das Betriebssystem Windows XP	21
2.1	Allgemeines	21
2.2	Der Startvorgang	21
2.2.1	Betriebssystemunabhängige Schritte	22
2.2.2	Betriebssystemabhängige Schritte	23
2.2.3	Der Anmeldevorgang	25
2.3	Architektur	26
2.4	Speicherverwaltung	28
2.5	Sicherheit in Windows Netzwerken	30
2.5.1	Sichere Authentifizierung im Netzwerk – Kerberos-Protokoll	30
2.5.2	Funktionsweise Kerberos	31
2.6	Die Registry	33
2.6.1	Aufbau und Struktur	33
2.6.2	Datentypen	35
2.7	Benutzerverwaltung mit dem Active Directory	35
2.7.1	Lokale Nutzerverwaltung – Registry-basierte Nutzerverwaltung	35

2.7.2	Active Directory – Datenbank-basierte Nutzerverwaltung	36
2.7.3	Zugriffsberechtigungen unter Windows – Das AGDLP Prinzip	39
	Literaturverzeichnis	43
	Stichwortverzeichnis	44

1 Firewalls

Definition

Eine Firewall besteht aus einer Gruppe von Netzwerkkomponenten (Hard- und Software) an der Schnittstelle zweier Netze. Sie gewährleistet die Einhaltung von Sicherheitsrichtlinien zwischen einem zu schützenden und einem unsicheren Netz (z. B. dem Internet). An dieser „Brandschutzmauer“ entscheidet sich, auf welche Dienste innerhalb des privaten Netzes zugegriffen werden kann und welche Dienste des nicht sicheren Netzes aus dem privaten Netz heraus nutzbar sind.

1.1 Funktion und Aufgaben von Firewall-Systemen

- Servicekontrolle
 - Netzwerkebene
 - Anwendungsebene
- Benutzerkontrolle
- Entkopplung von Diensten
- Verbergen des internen Netzes
- Protokollierung
- Alarmierung
- Zusatzfunktionen:
 - NAT¹
 - IPSec / VPN
 - Schutz vor Malware, etc.

¹Network Address Translation (NAT) ist in Rechnernetzen der Sammelbegriff für Verfahren, um automatisiert und transparent Adressinformationen in Datenpaketen durch andere zu ersetzen. Network Address Port Translation (NAPT) stellt mittlerweile die häufigste Form des NAT dar und wird daher oft als Synonym gebraucht. Da es neben der Umsetzung von IP-Adressen auch eine Umsetzung von Port-Nummern gestattet. Große Verbreitung fand NA(P)T durch die Knappheit öffentlicher IPv4-Adressen und die Tendenz, private Subnetze über Einwahlverbindungen mit dem Internet zu verbinden. Die einfachste Lösung des Problems beschränkter IP-Adressen war oft die durch NAT mögliche Verwendung mehrerer privater IP-Adressen mit nur einer öffentlichen IP-Adresse. Üblicherweise wird NAT an einem Übergang zwischen zwei Netzen durchgeführt. Der NAT-Dienst kann auf einem Router, einer Firewall oder einem anderen spezialisierten Gerät laufen. So kann zum Beispiel ein NAT-Gerät mit zwei Netzwerkadaptern das lokale private Netz mit dem Internet verbinden. Man unterscheidet zwischen Source NAT, bei dem die Quell-IP-Adresse ersetzt wird, und Destination NAT, bei dem die Ziel-IP-Adresse ersetzt wird. [1]

Ein Firewall stellt also einen klar definierten Zugang zu einem gesicherten Bereich dar. Dies entspricht in etwa der Aufgabe eines Pförtners bei einem Gebäude.

Ein Firewall bildet Netzabschnitte und schottet die zu schützenden Abschnitte ab, in dem er den einzigen sichern Übergang zwischen diesen Netzen darstellt. Dabei kontrolliert er wer Zugriff hat, über welche Protokolle oder Dienste zugegriffen wird, mit welchem Rechner kommuniziert werden darf und schreibt sicherheitsrelevante Vorfälle in ein Log.

1.2 Sicherheitspolitik

Eine Firewall ist mehr als nur eine Hardware oder Software. Damit effektiv Schutz geboten wird, muss eine Firewall:

- Auf einer Sicherheitspolitik aufsetzen,
- korrekt installiert und konfiguriert,
- korrekt administriert werden.

Dies stellt einen Kreislauf dar, der immer wieder ein Anpassen der Maßnahmen erfordert.

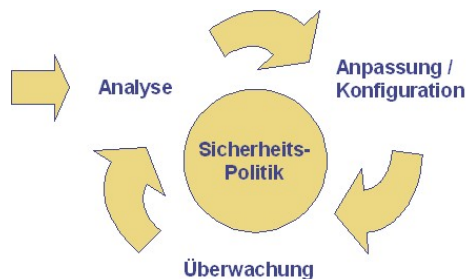


Abb. 1.1: *Sicherheitspolitik*

Um ein Konzept für die Sicherheitspolitik zu erstellen sind verschiedene Dinge zu klären bzw. festzulegen. Zum einem muss der Schutzbedarf geklärt werden, wobei auch bestimmte Vorgaben zu berücksichtigen sind. Desweiteren müssen die Anforderungen, wie Sicherheitsanforderungen und Kommunikationanforderungen festgelegt werden und Entscheidungen bezüglich der Organisation, Personal und Infrastruktur getroffen werden.

1.2.1 Sicherheitsanforderungen

Zu den Sicherheitsanforderungen gehören unter anderem:

- Zu schützende Ressourcen: Daten, Rechnersysteme, Kommunikationseinrichtungen, etc.
- Zugangskontrolle auf der Benutzerebene (Authentifizierung), Anwendungsebene, Netzwerkebene
- Verbergen der internen Netzstruktur
- Vertraulichkeit von Nachrichten

- Schutz gegen Angriffe auf Verfügbarkeit, z. B. für Informationsserver
- Schutz vor Angriffen durch das Bekannt werden von neuen sicherheitsrelevanten Softwareschwachstellen
- Anforderungen an das Firewall-System selber
- Behandlung von sicherheitsrelevanten Ereignissen

1.2.2 Kommunikationsanforderungen

Möglich Punkte:

- Diensten und Anwendungen
 - Unterscheidung von internen und externen Benutzer, ev. unterteilt nach Kommunikationsprofilen
 - Richtung der Dienste und Anwendungen
 - ggf. Anforderungen wie Authentisierungsverfahren, Verschlüsselung, Protokollierung, Zeitfenster
- Information, welche (nicht) nach aussen gelangen darf
- Filterregeln für die unteren Schichten (IP, ICMP, ARP, TCP und UDP) und für die Anwendungsschicht (SMTP, DNS, HTTP, etc.)
- Default Policy
- Verfügbarkeit
- Datendurchsatz

1.2.3 Maßnahmen bezüglich Organisation, Personal und Infrastruktur

Organisation

- Festlegung der Verantwortlichkeiten für Firewall-System (Sicherheitspolitik, Koordination, Umsetzung, Testen, Administration, etc.)
- Zugriffsrechte zum Security Management
- Kontrolle der Protokolldaten (wer, welche, wie oft, etc.)
- Reaktion auf Verletzungen der Sicherheitspolitik definieren
- Informationsbeschaffung zu Sicherheitslücken (u. a. des Firewall-Systems), Installation der Updates
- Betreuung der Benutzer
- Regelung für Wartungs- und Reparaturarbeiten

Personal

- Security Management
 - Profil des Security Administrators
 - Auswahl des Security Administrators und Externer
 - Vertreterregelung
 - Verfahren beim Ausscheiden eines Security Administrators
- Benutzer
 - Regelungen und Anweisungen

- Schulung der Benutzer
- Verfahren beim Ausscheiden eines Benutzers

Infrastruktur

- Sicherheitskritischen Zonen
- Sicherheitsmassnahmen und ihren Orte
- Firewall-Architektur
- Anforderung an von aussen erreichbare Systeme
- Netzzugänge (ISP, Modempool, etc.)
- Leitungsführung und physische Zugangssicherung

1.3 Typen von Firewalls

In der Praxis werden in der Regel zwei Typen von Firewalls eingesetzt und zwar *Paketfilter* Firewalls und *Application Gateways* (Proxys). Sie unterscheiden sich in der Art der Einbindung in das System und in ihrer Funktion.

1.3.1 Paketfilter

Ein Paketfilter analysiert und kontrolliert bis auf die *Transportebene*, d. h. er berücksichtigt beim Netzzugang den Absender, Empfänger und den Protokolltyp des Paketes. Entsprechend analysiert er im Netzwerk den IP- und ICMP- bzw. den TCP/UDP-Header. Nimmt man die Analogie Pförtner, entspräche das: „Der Pförtner prüft, ob das Logo auf dem LKW bekannt ist und lässt den Lastwagen passieren.“

1.3.2 Application Gateway (Proxy)

Ein Application Gateway analysiert und kontrolliert bis auf *Anwendungsebene*. Der Proxy entkoppelt das interne Netz logisch und physikalisch, d. h. es gibt keine direkte Verbindung zum zu schützenden Netz. Analogie zum Pförtner: „Der Pförtner prüft Papiere und Inhalt. Er nimmt die Pakete entgegen und bestellt einen Fahrer der eigenen Firma, der die Pakete zum eigentlichen Empfänger bringt.“

Ein Proxy arbeitet vorwiegend als aktiv in die Kommunikation eingreifende Vermittlungsstelle, welche Anfragen der einen Seite entgegennimmt, um dann eine eigene Verbindung zur anderen Seite aufzubauen. Er übernimmt somit stellvertretend für den Anfragenden (Client) die Kommunikation mit dem Ziel, wodurch eine Adressumsetzung realisiert wird. Während ein Proxy als separate Netzwerkkomponente auf diese Weise die wahre Absenderadresse eines der beiden Kommunikationspartner dem anderen Kommunikationspartner gegenüber komplett verbirgt, ist das bei einer lokal auf dem Quell- oder Zielsystem installierten Proxysoftware anders. Dort kann der Proxy lediglich den Port verschleiern. Ein aktiv in die Kommunikation eingreifender Proxy operiert auf der OSI-Schicht 7 und kann als vermeintlicher

Kommunikationspartner den Inhalt der Pakete zusammenhängend analysieren, dabei Anfragen filtern und bei Bedarf beliebige Anpassungen vornehmen, aber auch entscheiden ob und in welcher Form die Antwort des Ziels an den tatsächlichen Client weitergereicht wird. Mitunter dient er dazu, bestimmte Antworten zwischenspeichern, damit sie bei wiederkehrenden Anfragen schneller abrufbar sind, ohne sie erneut vom Ziel anfordern zu müssen. [2]

1.3.3 Gegenüberstellung Paketfilter – Application Gateway

Paketfilter	Application Gateway
<ul style="list-style-type: none"> • Gute Performance • Einfach erweiterungsfähig • Transparent • Verbergen i.A. zu schützendes Netz nicht • Daten oberhalb der Transportebene werden i.d.R. nicht analysiert 	<ul style="list-style-type: none"> • Sicherheitsfunktionen auf Anwendungsebene • Stellvertreter (Proxy) benötigt • Entkopplung der Dienste • Verbergen des internes Netzes • Bessere Protokollierungsmöglichkeiten • Geringe Flexibilität • Kosten i.d.R. höher

Tab. 1.1: Gegenüberstellung Paketfilter – Application Gateway

1.4 Komponenten einer Firewall

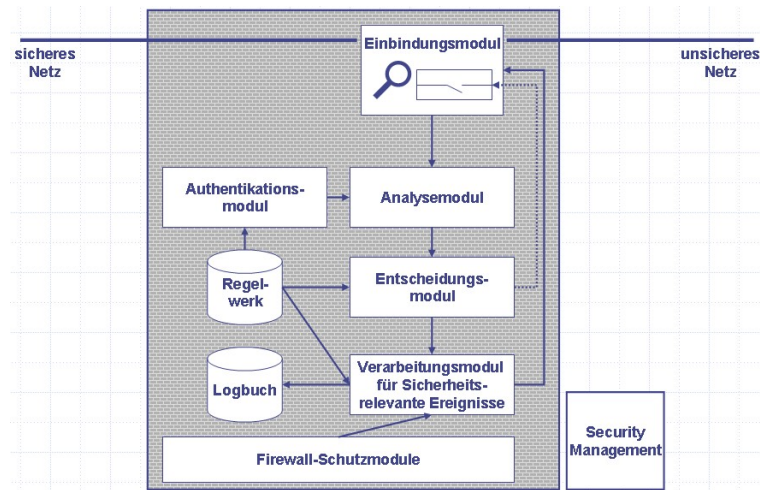


Abb. 1.2: Komponenten einer Firewall

- *Einbindungsmodul*: Realisiert Verbindung der Netze.
- *Analysemodul*: Analyse der Kommunikationsdaten. Paket Filter und Application Gateway analysieren auf unterschiedlichen Ebenen.
- *Entscheidungsmodul*: Auswertung der Analyseergebnisse. Vergleich mit den im Regelwerk festgelegten Definitionen der Sicherheitspolitik. Steuerung des Einbindungsmoduls.
- *Verarbeitungsmodul* für sicherheitsrelevante Ereignisse: Eintrag in das Logbuch und/oder Alarm, in Abhängig des Regelwerks.
- *Authentikationsmodul*: Identifikation und Authentisierung der Instanzen (Prozesse, Benutzer, etc.). Verschiedene Verfahren möglich.
- *Regelwerk*: Technische Umsetzung der Sicherheitspolitik und wird mit Hilfe eines Security Management erstellt.

- *Logbuch*: Enthält alle Protokolleinträge der sicherheitsrelevanten Ereignisse, die gemäss Regelwerk im Betrieb aufgezeichnet werden sollen.
- *Security Management*: Benötigt, um die Regeln für die Firewall festzulegen und die Protokolldaten aus den Logbuch zu analysieren.
- *Firewallschutzmodul*: Firewall-Element muss neben den Sicherheitsdiensten selber gegen Angriffe resistent sein.

1.5 Technische Implementierung

1.5.1 Schichtenmodell

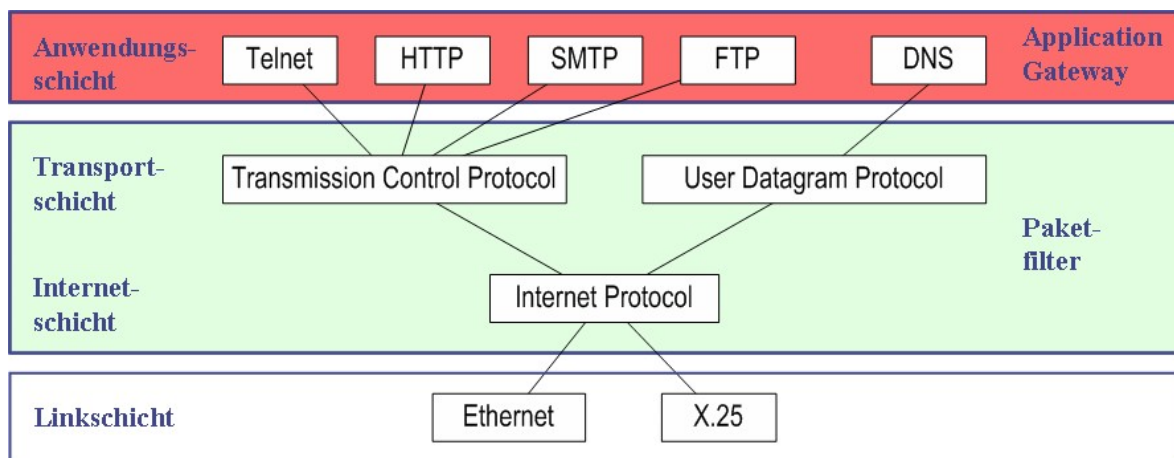


Abb. 1.3: Vereinfachtes OSI-Schichtenmodell

„Paketfilter“ meint grundsätzlich eine Filterung auf Ebene von Transport- und Internetschicht, während mit „Application Gateway“ eine Filterung im Bereich der Anwendungsschicht gemeint ist.

1.5.2 Der Paketfilter

Eine Filterung mit Hilfe eines Paketfilters findet aufgrund von sechs Eigenschaften statt:

- Pakete gehen *ein* oder *aus*.
- *IP-Adressen* von *Sender* und *Empfänger* sind aus der Internetschicht bekannt.
- *Portnummern* von *Sender* und *Empfänger* sind aus der Transportschicht bekannt.

Diese Eigenschaften können weiter verfeinert werden. So kann man *dynamische* Filterregeln erstellen, die nach einer gewissen Zeit wieder erlöschen oder man kann *zustandsabhängige* Filterregeln erstellen, die in Kraft treten, wenn ein bestimmter Zustand in einer höheren Schicht eintritt. Dynamische regeln setzt man zum Beispiel dann ein, wenn auf eine rausgehende anfrage eine Antwort erwartet wird, die den Paketfilter passieren soll. Ein Beispiel

für einen zustandsabhängigen Filter: Es wird versucht, auf den Zustand einer Verbindung zu schließen und dementsprechend zu filtern.

1.5.3 Application Gateway

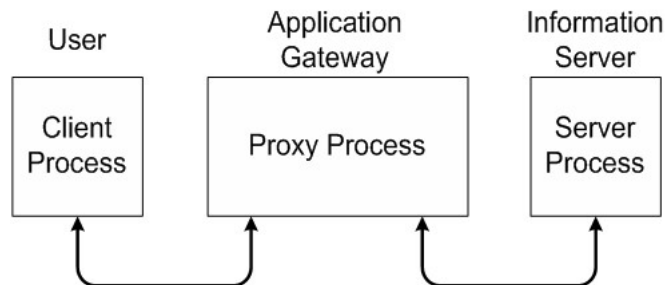


Abb. 1.4: Funktionsweise eines Application Gateways

Ein Application Gateway bzw. ein Proxy ist ein zwischengeschalteter Prozess, der meist auf einem eigenständigen Rechner läuft, der transparent vor das eigentliche Netzwerk geschaltet ist. Der Client im LAN „denkt“ der Proxy wäre der Server im Internet und der Server im Internet „denkt“, der Proxy wäre der Client für den die Daten bestimmt wären. Siehe dazu Grafik 1.4. Man unterscheidet dabei zwischen zwei unterschiedlichen Proxy Prozessen:

1. Application Level Proxy
2. Circuit Level Proxy

Ein *Application Level Proxy* besitzt je einen Prozess für ein Protokoll. Es werden alle Datenpakete ausgepackt, analysiert und entsprechend den Regeln behandelt. Zum Beispiel ist es so möglich einem HTTP-Proxy Pakete nach URLs zu Filtern oder nach Java Applets und so weiter. Dies ist mit einem Paketfilter nicht möglich. Ein *Circuit Level Proxy* ist ein generischer Prozess, der für alle Protokolle eingesetzt werden kann. Dafür braucht er entweder die protokollspezifischen Infos vom Client (=> dieser muss modifiziert werden, damit er das kann), oder er leitet die Verbindung einfach weiter, ohne überhaupt zu filtern.

1.5.4 Vergleich

Paketfilter

Schwächen:

- Filterregeln sind nur grobkörnig einstellbar.
- Bei vielen Regeln geht leicht die Übersicht verloren.

Stärken:

- Rel. einfache Konfiguration (neues Protokoll => neue Regeln)
- Effiziente Verarbeitung

Application Gateway

Schwächen:

- Jeder Prozess muss einzeln konfiguriert werden.
- Verarbeitung eher ineffizient

Stärken:

- Feinkörnige Filterung ist möglich.
- Inhaltliche Überprüfung ist möglich.
- Logische Trennung der Verbindung

Man sieht: Die Schwächen des einen Ansatzes sind die Stärken des anderen und umgekehrt.

1.6 Typische Netzwerk Topologien

1.6.1 Topologie nur mit Paketfilter

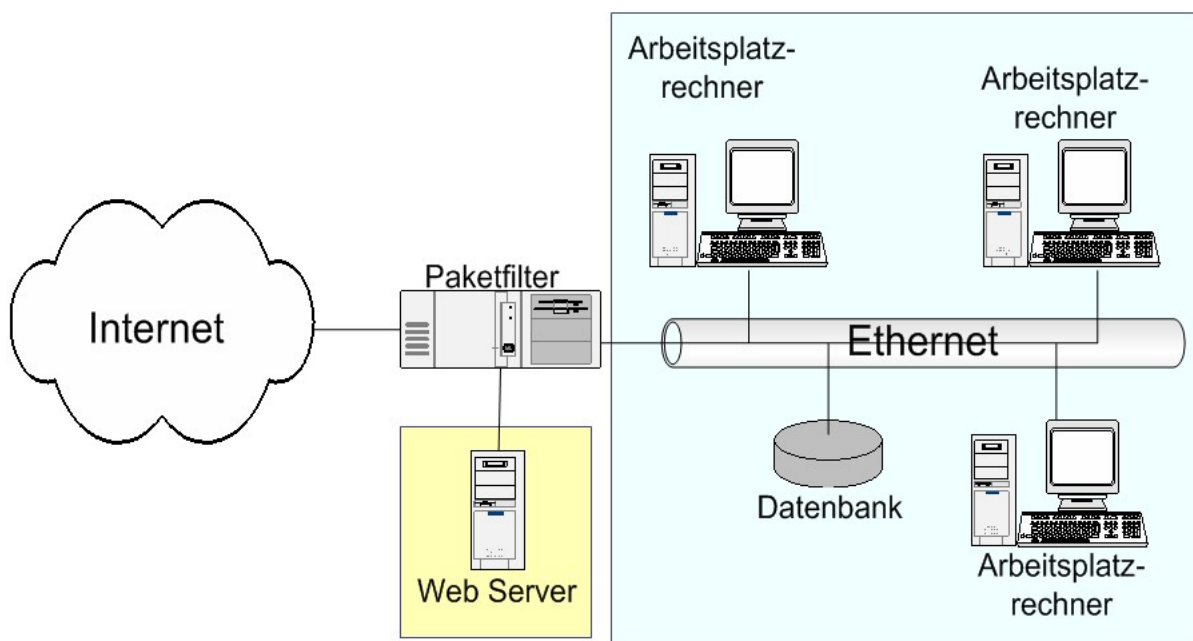


Abb. 1.5: Topologie nur mit Paketfilter

Einfachste (und billigste) Variante - ein einziger Paketfilter:

1. Aus dem Intranet heraus macht es keinen Sinn, nach IP Adressen zu filtern (kann leicht gefälscht werden)
2. Eine einzige Komponente trennt die beiden Netze => Konfiguration ist kritisch!
3. Erlaubte Dienste müssen beim End User abgesichert sein!

Die gezeichnete Variante (Abb. 1.5) enthält eine sog. „demilitarisierte Zone“, die hier gelb eingezeichnet ist. Die Idee dahinter ist, dass dort Dienste angeboten werden, die sowohl von innen als auch von aussen zugänglich sind ohne eine direkte Verbindung der zu trennenden Netze. Der Paketfilter kann hier Anfragen vom Internet an den Web Server weiterleiten, ohne dass das Intranet (hier blau) angesprochen werden muss. Das bedingt, dass der Paketfilter unterscheiden kann, von welchem Netzwerk-Interface er welche Anfrage erhalten hat. Zum jetzigen Zeitpunkt (Juli 2002) können das viele Produkte nicht!

1.6.2 Topologie nur mit Application Gateway

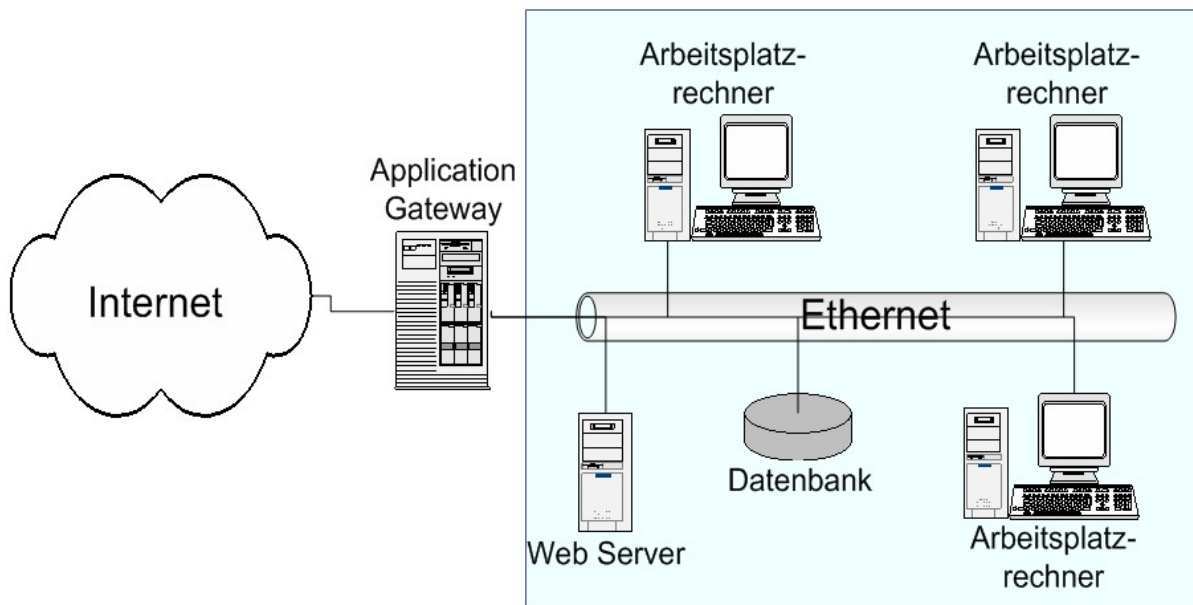


Abb. 1.6: Topologie nur mit Application Gateway

- Unterschied gegenüber vorheriger Lösung: Stärkerer Schutz, weil halt Application Gateways mehr schützen als nur Paketfilter
- Mit 2 oder noch mehr Interfaces wird nicht nur eine logische, sondern auch eine physikalische Trennung möglich.

Die hier gezeigte Lösung besitzt nur 2 Netzwerk-Interfaces und daher keine entmilitarisierte Zone. In diesem Beispiel wurde der Web Server im Intranet angehängt. Ein Zugriff auf den Web Server aus dem Internet führt als zu einer direkten Verbindung ins Intranet!

1.6.3 Screened Subnet Topologie mit Single Homed Application Gateway

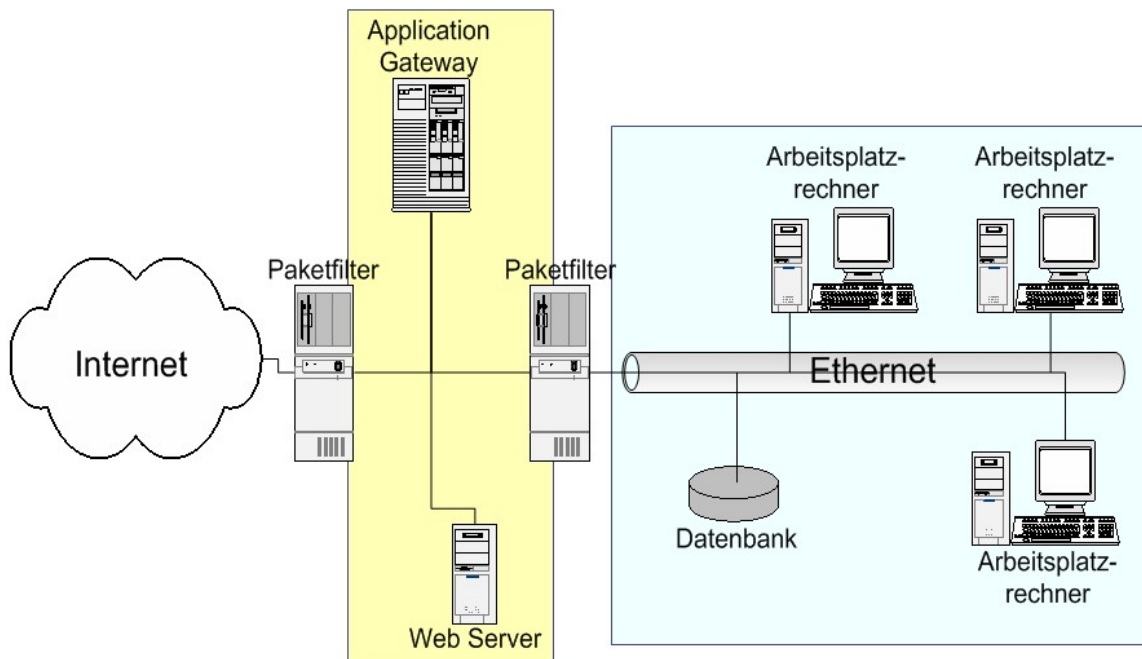


Abb. 1.7: Screened Subnet Topologie mit Single Homed Application Gateway

- Mehrere Schutzinstanzen => mehrere Hürden.
- Abstufung der Sicherheit in verschiedene Stärken ist möglich.
- Leicht skalierbar: ein zweiter Application Gateway lässt sich hier problemlos einfügen.
- Schwachstelle Information Server (hier Web Server): dieser kann gehackt und dann missbraucht werden für Angriffe gegen innen => Darum braucht es den 2. Paketfilter!

Wiederum in gelb gibt es hier eine demilitarisierte Zone, die in dieser Konfiguration als Screened Subnet bezeichnet wird. Der Application Gateway besitzt in diesem Fall nur ein einziges Netzwerk-Interface (single homed), so dass die beiden Paketfilter erzwingen müssen, dass ein- und ausgehende Datenpakete über den Application Gateway laufen.

1.6.4 Screened Subnet Topologie mit Multi Homed Application Gateway

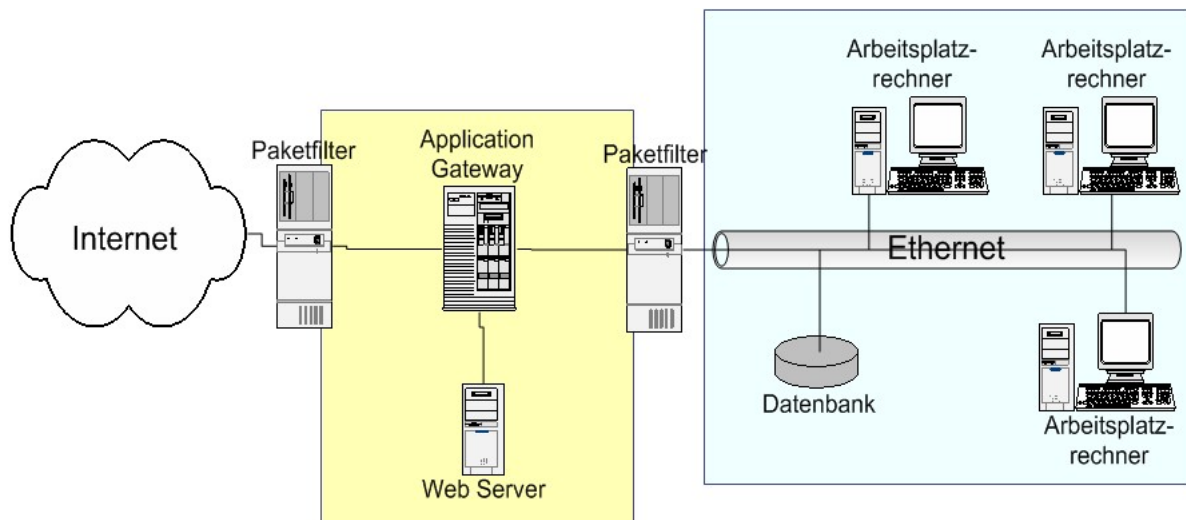


Abb. 1.8: Screened Subnet Topologie mit Single Homed Application Gateway

- Symmetrische Lösung => gut für ungewisse Zustände innerhalb des Intranets, Bsp. Provider, Universität.
- Wenig Flexibilität: im Idealfall ist jedes Kästchen ein Produkt eines anderen Herstellers (damit nicht alle die gleichen Sicherheitslücken haben), d.h. ich habe drei verschiedene Produkte, die ich warten muss.

1.6.5 Beliebiges Konzept

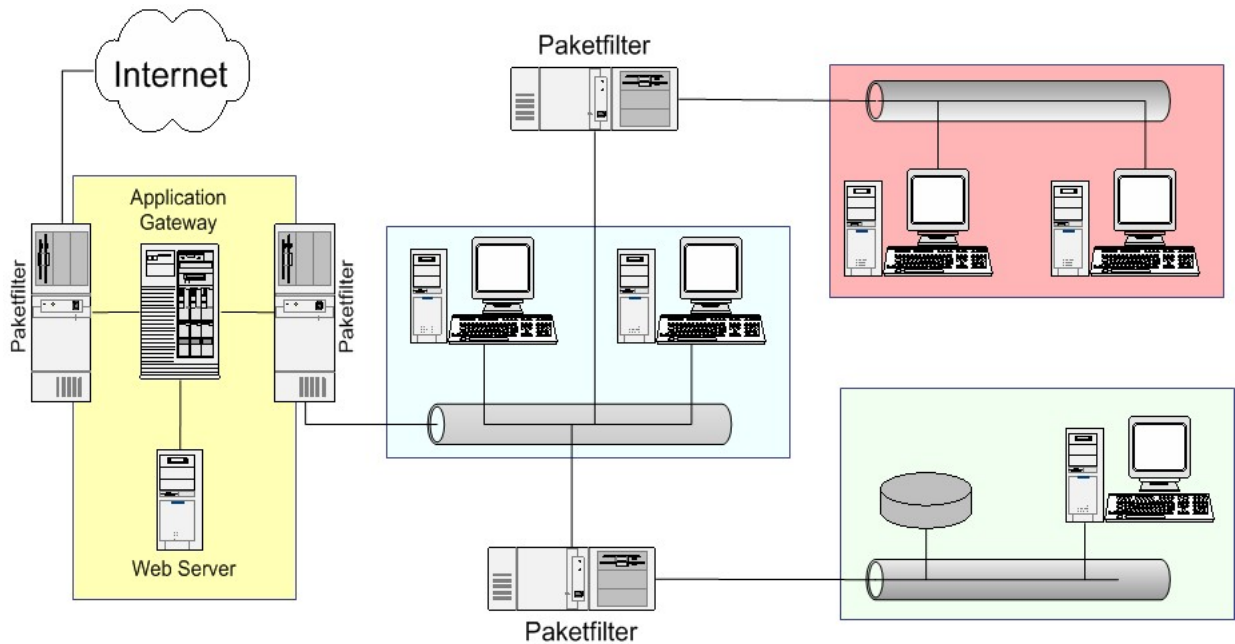


Abb. 1.9: *Beliebiges Konzept*

- Verschiedene Zonen, anpassbar an die Bedürfnisse der einzelnen Teile des Intranets. Bsp.: Grüne Zone (Buchhaltung) soll keinen Zugang haben zu den Daten der roten Zone (Forschung und Entwicklung).
- Im Beispiel 5 Maschinen, im Idealfall alles unterschiedliche Produkte => enormer Wartungsaufwand!

Die gezeichnete Konfiguration ist nur ein Beispiel. Es wäre genauso denkbar, die rote Zone direkt am zweiten Paketfilter von links anzuhängen, oder die blaue Zone zwischen die rote, oder ... je nach den Bedürfnissen.

2 Das Betriebssystem Windows XP

Ein Betriebssystem ist die Software, die die Verwendung (den Betrieb) eines Computers ermöglicht. Es verwaltet Betriebsmittel wie Speicher, Ein- und Ausgabegeräte und steuert die Ausführung von Programmen. Das Betriebssystem stellt die Schnittstelle zwischen dem Benutzer und der Hardware des Computers dar.

2.1 Allgemeines

Windows XP (interner Codename in der Entwicklungsphase: Whistler) ist ein Betriebssystem der Firma Microsoft. *Windows XP* (*Windows NT* Version 5.1) kam am 25. Oktober 2001 auf den Markt und ist der technische Nachfolger von *Windows 2000* (*Windows NT* Version 5.0) mit *Windows-NT*-Betriebssystemkern. Zusätzlich löste es *Windows Me* der *MS-DOS*-Linie in der Version *Home Edition* als Produkt für Heimanwender und Privatnutzer ab. Die *MS-DOS*-Linie wurde daraufhin von Microsoft eingestellt.

Windows XP ist, wie seine Vorgänger ein Mehrbenutzerbetriebssystem. Ein Mehrbenutzerbetriebssystem oder Multiuser-System ist ein Betriebssystem, das die Fähigkeit hat, Arbeitsumgebungen für verschiedene Benutzer bereitzustellen und voneinander abgrenzen zu können. Desweiteren unterstützt *Windows XP* präemptives Multitasking¹. Multitasking bezeichnet die Fähigkeit eines Betriebssystems, mehrere Aufgaben (Tasks) nebenläufig auszuführen. Dabei werden die verschiedenen Prozesse in so kurzen Abständen immer abwechselnd aktiviert, dass der Eindruck der Gleichzeitigkeit entsteht.

2.2 Der Startvorgang

Um einen PC zu starten müssen gewisse Mindestanforderungen gegeben sein. Es muss mindestens ein Motherboard mit CPU, Arbeitsspeicher und einer Grafikkarte vorhanden sein.

¹Im Gegensatz zum kooperativen Multitasking entscheidet das Betriebssystem über die Zuteilung von CPU Rechenzeit und nicht der laufende Prozess.

2.2.1 Betriebssystemunabhängige Schritte

Für die betriebssystemunabhängigen Schritte des Startvorganges ist das BIOS² verantwortlich. Im wesentlichen führt das BIOS folgende Schritte aus bevor das Betriebssystem gestartet wird:

1. Durchführen des *Power on self Tests* (POST). Der Power On Self Test (POST) ist ein Selbsttest nach dem Einschalten des Computers – ein Vorgang, den der Computer beim Hochfahren durchläuft, um zu prüfen, ob die grundlegenden Komponenten des PCs funktionsfähig sind.
2. Initialisierung der Hardware.
3. Aufrufen von BIOS-Erweiterungen, die auf Steckkarten untergebracht sind.
4. Feststellen, von welchem Datenträger gebootet werden kann und soll. Dies geschieht durch Lesen der Spur 0 der an dem primären IDE Controller angeschlossenen Festplatte. Dort befindet sich die Partitionstabelle mit Informationen über die Größe der Partitionen mit Start- und Endzylinder. Ziel ist es die Systempartition zu finden.
5. Nach dem die Systempartition gefunden wurde, wird der Bootloader am Anfang der Partition geladen. Der Bootloader lädt dann weitere Teile des Betriebssystems.

Bei klassischen im Real-Mode³ laufenden Betriebssystemen (z. B. DOS) wird das BIOS auch im weiteren Betrieb genutzt. Es übernimmt für das Betriebssystem die Kommunikation mit diverser Hardware, z. B.:

- Tastatur
- serielle und parallele Schnittstellen
- Systemlautsprecher
- Grafikkarte
- Diskettenlaufwerke
- Festplatten

Moderne Arten von Hardware werden vom BIOS nicht bedient. Zur Ansteuerung z. B. einer Maus ist unter DOS ein spezieller Hardwaretreiber nötig. Neuere, treiberbasierte Betriebssysteme wie z. B. Linux oder Microsoft Windows nutzen diese BIOS-Dienste nicht. Sie laden für jede Art von Hardware einen speziellen Treiber. Sie müssen jedoch am Anfang ihres Startvorgangs noch kurz auf die BIOS-Funktionen zur Ansteuerung der Festplatten zurückgreifen, um ihren Festplattentreiber zu laden.

Bis zu diesem Punkt läuft der Prozessor noch im Real-Mode.

²BIOS = Basic Input Output System. Das BIOS ist ein Programm, das in einem nichtflüchtigen Speicher auf der Hauptplatine des PCs abgelegt ist und das unmittelbar nach dessen Einschalten zur Ausführung gelangt. Aufgabe des BIOS ist es, den PC soweit zu starten, bis das eigentliche Betriebssystem die Kontrolle übernehmen kann.

³Im Real Mode benutzt der Prozessor die gleiche Methodik wie der Intel 8086-Prozessor, um auf den Hauptspeicher zuzugreifen.

2.2.2 Betriebssystemabhängige Schritte

Der Bootmanager von Windows heißt *NTLDR*. Dieser schaltet in den 32 Bit-Modus und lädt von der Start-Partition grundlegende Funktionen zum Zugriff auf das Dateisystem (z. B. Treiber, Controller und die Unterstützung für NTFS). Im Anschluss wird die *boot.ini* ausgelesen und falls erforderlich das Bootmenü angezeigt. Startet man hier Win9x/ME/DOS wird die Datei *bootsec.dos* ausgewertet. (Sie Enthält den Code des 9x/ME/DOS Bootsectors. Die Datei kann auch einen beliebigen anderen Namen haben z. B. Boot.w98). Startet man hier WinXP/NT/W2k startet *ntdetect.com* die Hardwareerkennung. Hier werden Erkenntnisse über wichtigen Informationen zum Systemfirmenware, Bustypen, Grafikkarte, Tastatur, Schnittstellen, Maus, Disklaufwerke, Festplatten und die aktuelle Systemzeit gesammelt. Weiterhin werden hier die vorhandenen ISA-Plug and Play Karten erkannt. Bei APIC⁴ fähigen System spielt *ntdetect.com* eine nicht so große Rolle. Die Ressourcen (IRQ⁵, Adressen) werden von WinXP verwaltet. Beim System ohne APIC werden von *ntdetect.com* die Ressourcen (IRQ, Adressen) von BIOS übernommen und an NTLDR übermittelt.

Anschließend startet NTLDR den Betriebssystem-Kernel *ntoskrnl.exe* und die Hardware Abstraction Layer *hal.dll*⁶. Diese bilden die Windows-Ausführungsschicht, die Konfigurationsinformationen verarbeitet, Treiber und Dienste lädt.

Anschließend wird die Registry geladen. Beim Start wird der Zweig HKLM⁷\SYSTEM ausgelesen. Hier werden Sätze von Hardwarekonfigurationen verwaltet (ControlSet001, ControlSet002 etc.). Der Eintrag CurrentControlSet ist eine Kopie des aktuell verwendeten Konfigurationsdatensatzes. Welcher Konfigurationsdatensatz im welchen Fall verwendet werden soll, wird im Registry im Schlüssel HKEY_LOCAL_MACHINE\SYSTEM>Select festgelegt. Der NTLDR lädt den festgelegten Konfigurationsdatensatz. Zu diesen Konfigurationsdaten werden noch die von *ntdetect.com* übergebenen Hardwareinformationen verwendet und der HKLM\Hardware angelegt. *ntoskrnl.exe* sucht jetzt Treiber und Dienste die gestartet werden müssen anhand HKLM\SYSTEM\CurrentControlSet\Services. Jetzt startet der Sitzungs-Manager *smss.exe*. Dieser erstellt die Systemumgebungsvariablen und startet den Kernel-Modusabschnitt des Win-Subsystems %systemroot%\system32\win32k.sys (Die benutzerspezifischen Umgebungsvariablen werden erst nach Benutzeranmeldung gesetzt). *win32k.sys* initialisiert den Grafik-Modus von XP (ab hier Grafische-Oberfläche).

Bis zu diesem Punkt läuft alles im Kernel-Mode ab. Jetzt wird der Benutzermodusabschnitt des Win-Subsystems geladen: %systemroot%\system32\csrss.exe. Ausgewertet wird die Liste aus

HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\SubSystems. Zu diesem Zeitpunkt wird auch die Auslagerungsdatei (*pagefile.sys*) angelegt. Das Win-Subsystems *csr-*

⁴Der Advanced Programmable Interrupt Controller (APIC, nicht zu verwechseln mit ACPI) sorgt für die Verteilung von Interrupts in x86- und Itanium-basierenden Computersystemen.

⁵Eine Unterbrechungsanforderung oder englisch Interrupt Request (IRQ) löst eine Unterbrechung der Prozessbearbeitung eines Prozessors im System aus. In der Regel wird diese von Geräten im System durch eine Busleitung mit Namen IRQ signalisiert und der Prozessor reagiert mit einer Programmumschaltung oder englisch „context switch“ und führt die Unterbrechungsroutine aus. Nach deren Beendigung wird IRQ zurückgesetzt und die unterbrochene Aufgabe fortgesetzt.

⁶Die HAL (Hardware Abstraction Layer) dient so als Schnittstelle für die meisten Betriebssystemprozesse und übernimmt die Kommunikation mit der Hardware.

⁷HKLM: HKEY_LOCAL_MACHINE

ss.exe und alle davon gestarteten Komponenten der Benutzeroberfläche laufen in User-Mode (Prozesse können nicht direkt auf Hardware zugreifen).

Nachdem die Treiber und Dienste gestartet sind, startet das Windows-Subsystem den Anmelde Dienst *winlogon.exe*. Dieser Prozess startet *services.exe* (Dienststeuerungs-Manager) und *lsass.exe* (die lokale Sicherheitsautorität). Bei der schnellen Benutzeranmeldung werden sofort die Sicherheits- und Authentifizierungskomponenten installiert, sodass der Benutzer beim Anklicken seines Namens sofort identifiziert werden kann. Dieses Verfahren ist neu und erst in WinXP erhalten. Wurde die schnelle Benutzeranmeldung deaktiviert, geschieht dies erst nach der Benutzeranmeldung. Nach der Benutzeranmeldung wird die letzte als funktionierend bekannte Konfiguration mit der aktuell funktionierenden Konfiguration überschrieben, dann liest das System die Gruppenrichtlinien, die für den angemeldeten Benutzer gelten, und konfiguriert damit Zugriffsrechte und Einstellungen der Benutzeroberfläche. Die Kommunikation zwischen der *winlogon.exe* und der *lsass.exe* erfolgt über die *msgina.dll*⁸. Winlogon kann für die Verwendung einer anderen GINA-Bibliothek konfiguriert werden, die andere Authentifizierungsmethoden wie z. B. Smartcards oder Fingerabdruckscanner unterstützen, oder eine andere Anmeldeoberfläche bereitstellen.

Jetzt startet WinXP alle automatisch auszuführenden Programme, die in den entsprechenden Registryzweigen und in der Autostartgruppe des Startmenüs liegen.

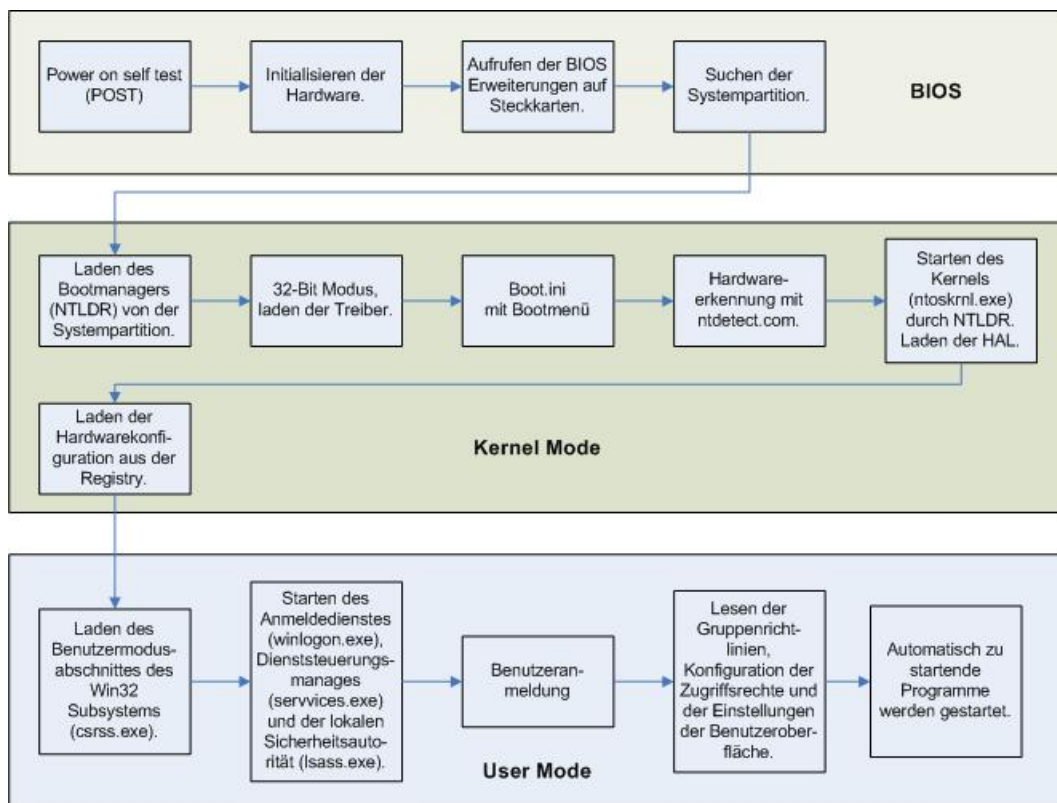


Abb. 2.1: Übersicht Startvorgang Windows XP

⁸GINA steht für Graphical Identification And Authentication, sie stellt den sicheren Authentifizierungs- und interaktive Anmelde Dienste zur Verfügung.

2.2.3 Der Anmeldevorgang

Für den Anmeldevorgang sind unter Windows im wesentlichen zwei Prozesse/Dateien verantwortlich. Das sind zum einem die *WinLogon.exe* und die *MSGina.dll*. Der Prozess *Winlogon* ist verantwortlich für den interaktive Anmeldevorgang. Er erzeugt einen neuen Desktop und zeigt eine Benutzerschnittstelle an, in der der Benutzer seine Zugangsdaten eingeben kann. Die *MSGina.dll* beinhaltet alle Benutzerschnittstellen Komponenten, die vom *Winlogon* Prozess angezeigt werden. Betrachtet man die Ressourcen der *msgina.dll*, findet man viele bekannte Dialoge. Sie dazu die Grafiken 2.2 (S. 25), 2.3 (S. 25) und 2.4 (S. 25).

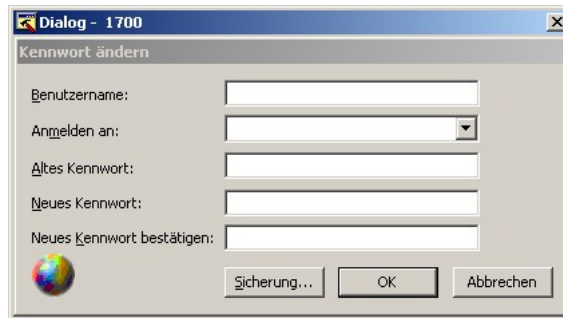


Abb. 2.2: Dialog zum Kennwortändern



Abb. 2.3: Sicherheitsdialog

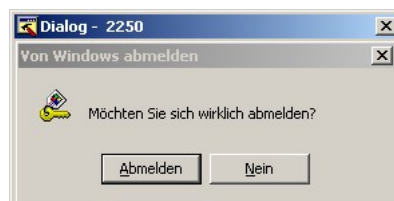


Abb. 2.4: Sicherheitsabfrage beim Abmelden

Die GINA macht alles was mit der Authentifizierung eines Benutzers zusammenhängt. Dazu zählt als sichtbarste Maßnahme der Logon-Bildschirm mit der Logonbox, welche in der sogenannten SAS (Secure Attention Sequence) laufen. Dazu zählt im übrigen auch der „Bildschirm“ der erscheint, wenn man Strg+Alt+Entf drückt. Genau das ist auch der Grund, warum

man beispielsweise die GINA ersetzen kann um Strg+Alt+Entf zu unterdrücken, aber ohne einen Tastaturfiltertreiber schreiben zu müssen. Die GINA erstellt beim Einloggen eines Benutzers über die LSA (Local Security Authority), nachdem die Identität des Benutzers überprüft wurde, eine sogenannte Logon Session. Eine Logon Session ist etwas imaginäres, was durch ein Token quasi materialisiert wird. Durch das Token eines eingeloggten Benutzers (jeder Prozess und diverse andere Objekte besitzen dieses Token) wird der Benutzer identifiziert sobald er sich einmal eingeloggt hat (bei einem Zugriff auf eine Ressource muss nicht erneut nach dem Passwort gefragt werden, sondern das Token symbolisiert den Benutzer). Nachdem die Logon Session erzeugt wurde, startet Winlogon den Hilfsprozess Userinit.exe. Dieser Prozess stellt die Benutzerumgebung wieder her, stellt eventuelle Netzwerkverbindungen wieder her, führt Loginskripte aus und startet die Shell.

2.3 Architektur

Windows XP basiert komplett auf der Struktur von Windows 2000. XP ist modular aufgebaut, das heißt jede Systemfunktion und jedes Subsystem wird von einem Modul oder einer kleinen Gruppe von Modulen bedient. Der Vorteil dieses Designs liegt auf der Hand - Fehlerhafte Module lassen sich leicht austauschen und neue Funktionen leicht integrieren. Das modulare Design hat einen weiteren entscheidenden Vorteil – Portabilität. Alle Hardware-spezifischen Funktionen sind im sogenannten Hardware Abstraktion Layer (HAL) zusammengefasst, der die Vermittlungsschicht zwischen Betriebssystem und Hardware bildet. Um Windows XP an andere Plattformen anzupassen muss lediglich für den HAL neuer Code geschrieben werden. Die restlichen Komponenten werden einfach neu kompiliert.

Windows XP unterscheidet zwischen dem so genannten User- und dem Kernel-Mode. Module im Kernel-Mode haben beispielsweise direkten Zugriff auf die Hardware oder den Speicher. Das ermöglicht eine höhere Performance. Im Gegenzug steigt aber auch das Risiko eines fehlerhaften Speicherzugriffes. Module im User-Mode sind komplett von der Hardware abgeschottet und können Systemfunktionen nur über die so genannten Executive Services ausführen. Die Executive Services sind eine Sammlung von Komponenten, die den Zugriff auf Hardware und Ressourcen verwalten.

Im einzelnen sind das:

- *I/O Manager*: Ist zuständig für die Organisation von Ein- und Ausgabe auf verschiedene Geräte.
- *Filesystem-Manager*: Eine Unterfunktion des I/O-Managers ist der Filesystem-Manager, der Zugriffe auf Speichermedien wie Festplatten, Bandlaufwerke oder Netzwerk-Freigaben verwaltet.
- *IPC Manager*: Verarbeitet die gesamte Kommunikation zwischen verschiedenen Prozessen. Diese Kommunikation kann lokal über den LPC (Lokal Procedure Call) erfolgen oder mit Prozessen auf anderen Rechnern via RPC (Remote Procedure Call).
- *Memory Manager*: Für die wichtigste Ressource im Rechner, den Speicher, ist eine eigene Komponente verantwortlich. Der Speichermanager stellt jedem Prozess seinen eigenen virtuellen Adressraum zur Verfügung und sichert die verschiedenen Adressräume voneinander ab.

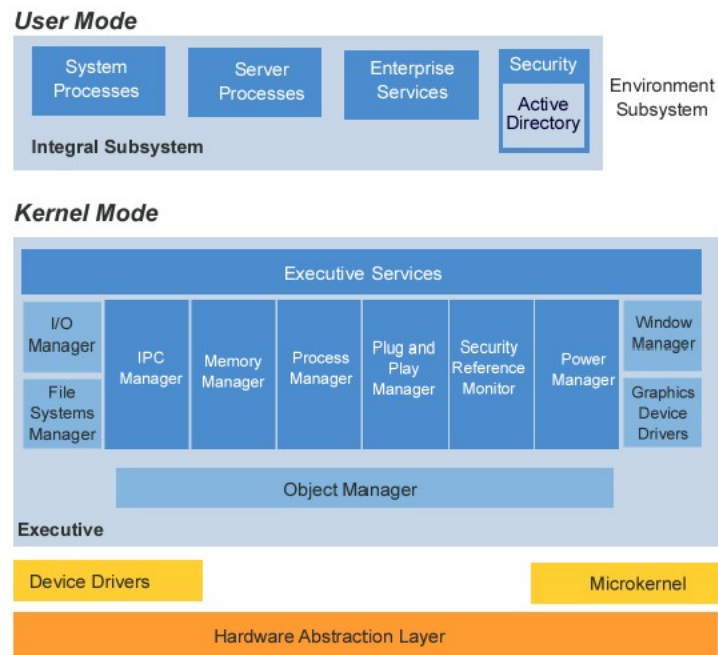


Abb. 2.5: Systemarchitektur von Windows XP

- *Process Manager*: Verwaltet und überwacht alle im System ablaufenden Prozesse.
- *Plug and Play Manager*: Ist für die Erkennung und Überwachung von installierten PnP-Geräten zuständig und handhabt die Installation von Treibern sowie das Starten notwendiger Dienste.
- *Security Reference Monitor*: Überwacht alle Sicherheitsmechanismen wie Authentifizierung, Zugriffe oder Besitzrechte.
- *Power Manager*: Zuständig für alle Funktionen des Power-Managements in Windows XP, wie Batterieüberwachung oder Stromsparfunktionen.
- *Window Manager*: Verwaltet die Benutzerschnittstelle wie etwa Dialogboxen, Fenster oder Benutzereingaben.
- *Graphics Device Drivers*: Sind zuständig für die eigentliche Ausgabe der Informationen auf dem Monitor.
- *Object Manager*: Alles in Windows XP wird als Objekt verwaltet. Dementsprechend ist der Object Manager eine zentrale Instanz von Windows XP.

Der Mikrokernel⁹ von Windows ist die zentrale Schaltstelle des Betriebssystems. Er verwaltet die Ausführung auf dem Prozessor und die Hardware-Interrupts. Zudem koordiniert er alle Aktivitäten der Executive Services.

[3]

⁹Der Mikrokernel verfügt im Gegensatz zu einem monolithischen Kernel nur über grundlegende Funktionen – in der Regel lediglich Funktionen zur Speicher- und Prozessverwaltung, sowie Grundfunktionen zur Synchronisation und Kommunikation.

2.4 Speicherverwaltung

Früher war es bei älteren Betriebssystemen so, dass der physische Speicher gleichbedeutend mit dem im Rechner vorhandenen Arbeitsspeicher war. Das heißt, waren nur 16 MB Speicher eingebaut, so hatten alle Prozesse auch nur 16 MB zur Verfügung. Moderne Betriebssysteme arbeiten mit einem so genannten virtuellen Adressraum. Das heißt, jeder Prozess bekommt einen Adressraum fester Größe zugewiesen, unter Windows sind dies vier Gigabyte. Wobei ein Prozess aber für sich selber nur zwei Gigabyte nutzen kann, die restlichen zwei Gigabyte sind für das Betriebssystem reserviert. Dorthin werden die Betriebssystemfunktionen aus den Systemdateien gemappt. Dies ist erforderlich, da die Adressräume der Prozesse vom Betriebssystem untereinander abgeschottet werden, so dass ein Prozess nicht auf den Adressraum eines anderen Prozesses ohne weiteres zugreifen kann. Dies trägt erheblich zu Stabilität des Systems bei, da ein Fehler in einem Prozess sich nicht auf einen anderen Prozess auswirken kann. Da die Betriebssystemfunktionen in seinen Adressraum gemappt werden, kann er auf selbige zugreifen.

Nun ist es aber so, dass der Hauptspeicher nicht groß genug ist, dass allen laufenden Prozessen ein vier Gigabyte großer Adressraum zugewiesen werden könnte. Hier kommt der Begriff *virtuell* ins Spiel. Das Betriebssystem nutzt eine Datei auf der Festplatte als Erweiterung für den Arbeitsspeicher. Dies ist die so genannte Auslagerungsdatei oder auf Englisch Pagefile. Der Prozess „denkt“, er hätte einen Adressraum von vier Gigabyte zur Verfügung auf den er ohne weiteres zugreifen könnte. Dies ist aber nicht der Fall. Tatsächlich befinden sich nur die Daten im Arbeitsspeicher, die gerade gebraucht werden.

Für die Verwaltung dieses virtuellen Speichers ist natürlich eine erhebliche Unterstützung der CPU erforderlich. Wenn ein Thread versucht auf ein Byte zuzugreifen, muss die CPU natürlich wissen, wo sie dieses Byte findet, im Arbeitsspeicher oder in der Auslagerungsdatei. Es gibt also zwei Möglichkeiten, die angeforderten Daten befinden sich im Arbeitsspeicher und können direkt verwendet werden oder eben nicht und müssen erst in den Arbeitsspeicher geschrieben werden, damit die CPU darauf zugreifen kann. Was genau passiert, wenn ein Thread auf einen Datenblock zugreift, verdeutlicht folgendes Schema 2.6 auf Seite 29.

Die erste Möglichkeit ist natürlich trivial und die günstigste, der Datenblock befindet sich bereits im Arbeitsspeicher. In diesem Fall bildet die CPU lediglich die virtuelle Adresse der Daten auf die physische Adresse im RAM ab und ermöglicht so den gewünschten Zugriff.

Allerdings kommt es jedoch öfters vor, dass sich die gewünschten Daten anstatt im RAM, irgendwo in der Auslagerungsdatei auf der Festplatte befinden. Ein Zugriff auf die Daten erzeugt dann einen so genannten Seitenfehler. Diese Seitenfehler kann man sich auch mal im Taskmanager ansehen, siehe Grafik 2.7 auf Seite 30.

Mit diesen Seitenfehlern informiert die CPU das Betriebssystem darüber, dass ein Zugriffsversuch misslungen ist. Das Betriebssystem guckt daraufhin, ob noch eine freie Seite¹⁰ im Arbeitsspeicher zur Verfügung steht. Steht eine zur Verfügung, werden die angeforderten Daten dorthin geladen und die CPU kann darauf zugreifen. Wird keine freie Speicherseite gefunden, muss eine belegte Seite freigegeben werden. Wurde die Seite nicht verändert,

¹⁰Eine Seite ist die Grundeinheit, die bei der Speicherverwaltung verwendet wird.

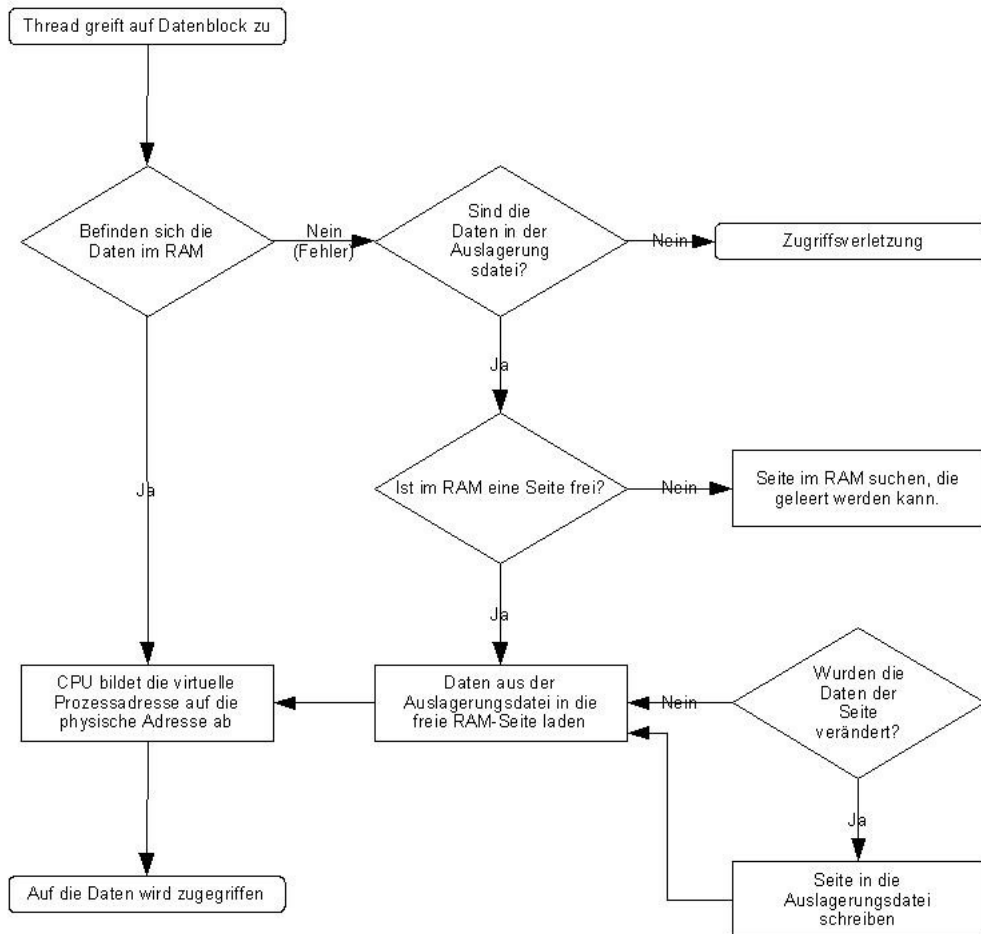


Abb. 2.6: Schema Speicherzugriff

kann sie sofort freigegeben werden. Wurde sie aber verändert, muss sie zuvor in die Auslagerungsdatei kopiert werden. Dann greift das System auf die Auslagerungsdatei zu, sucht die angeforderten Daten und kopiert sie in den Arbeitsspeicher. Ist der Vorgang abgeschlossen, aktualisiert das Betriebssystem seine Tabelle und zeigt an, dass die virtuelle Speicheradresse jetzt auf die passende Adresse des physikalischen Speichers im RAM zeigt. Die CPU probiert jetzt erneut die Anweisung aus, die ursprünglich den Seitenfehler auslöste. Dies mal aber findet die CPU die Daten im RAM und kann auf die Daten zugreifen.

Es dürfte klar sein, dass je öfter das Betriebssystem diesen Vorgang wiederholen muss, desto langsamer wird es. Denn anstatt zum Beispiel das zu startende Programm in den Arbeitsspeicher zu laden, ist es damit beschäftigt Speicherseiten aus- und einzulagern. Man kann dieses Verhalten aber verbessern, in dem man dem System entsprechen mehr Arbeitsspeicher zur Verfügung stellt, damit nicht mehr so viel ein- und ausgelagert werden muss. Meist ist der Leistungsgewinn durch mehr RAM größer, als der, der sich durch das Ersetzen einer schnelleren CPU erreichen läßt.

[9]



Name	Benutzername	C...	Speichera...	Seitenfehler
mdm.exe	SYSTEM	00	2.376 K	640
taskmgr.exe	Michael	00	4.308 K	1.108
ConTEXT.exe	Michael	00	11.536 K	3.003
alg.exe	LOKALER DIENST	00	3.308 K	867
explorer.exe	Michael	00	21.892 K	49.240
taskmgr.exe	Michael	00	27.648 K	1.108

Abb. 2.7: Seitenfehler im Windows Taskmanager

2.5 Sicherheit in Windows Netzwerken

2.5.1 Sichere Authentifizierung im Netzwerk – Kerberos-Protokoll

Jeder Benutzer einer XP-Domäne muss über ein Benutzerkonto im Active Directory verfügen. Mit einem entsprechenden Passwort meldet sich der Benutzer über das Kerberos¹¹-Protokoll (Version 5) an der Domäne an. Mit diesem Protokoll wurden die bekannten Sicherheitsdefizite des NTLM-Protokolls von Windows NT 4.0 beseitigt und damit ein wesentlicher Sicherheitsgewinn erzielt.

Das vom MIT (Massachusetts Institute of Technology) entwickelte Kerberos-Protokoll garantiert zudem Kompatibilität mit anderen Kerberos-basierten Systemen, etwa UNIX. Das wesentliche Prinzip der Kerberos-Authentifizierung besteht in der Verlagerung der Identitätsprüfung auf einen zentralen Dienst. Dieser verwaltet in einer Datenbank alle Ressourcen (u. a. Benutzer, Rechner, Netzwerkdienste) mit ihren jeweiligen Schlüsseln. Bei Benutzern werden diese Schlüssel von den Benutzer-Passwörtern abgeleitet.

Kerberos ist ein verteilter Authentifizierungsdienst (Netzwerkprotokoll), der für offene und unsichere Computernetze (zum Beispiel das Internet) von Steve Miller und Clifford Neuman basierend auf dem Needham-Schroeder-Protokoll zur Authentifizierung (1978) entwickelt wurde. Die zurzeit aktuelle Version ist Kerberos 5. Sie ist in RFC 4120 definiert und nutzt ASN.1 zur Codierung.

Kerberos bietet sichere und einheitliche Authentifizierung in einem ungesicherten TCP/IP-Netzwerk auf sicheren Hostrechnern. Die Authentifizierung übernimmt eine vertrauenswürdige dritte Partei. Diese dritte Partei ist ein besonders geschützter Kerberos-5-Netzwerkdienst. Kerberos unterstützt Single Sign On, das heißt, ein Benutzer muss sich nur noch einmal anmelden, dann kann er alle Netzwerkdienste nutzen, ohne ein weiteres Mal ein Passwort eingeben zu müssen. Kerberos übernimmt die weitere Authentifizierung.

Das Kerberos-Protokoll stellt auch sicher, dass zu keinem Zeitpunkt offene oder verschlüsselte Passwörter übertragen werden.

¹¹Kerberos ist in der Griechischen Mythologie der dreiköpfige Wachhund an den Pforten zur Unterwelt

2.5.2 Funktionsweise Kerberos

Bei Kerberos sind drei Parteien beteiligt: der Client, der Server, den der Client nutzen will, und der Kerberos-Server.

Der Kerberos-Dienst authentisiert sowohl den Server gegenüber dem Client, als auch den Client gegenüber dem Server, um Man-In-The-Middle-Angriffe zu unterbinden. Auch der Kerberos-Server selbst authentisiert sich gegenüber dem Client und Server und verifiziert selbst deren Identität.

Kerberos verwendet sog. Tickets zur Authentifizierung. Um den Kerberos-Dienst nutzen zu können, muss sich ein Client zuerst beim Kerberos-Server anmelden. Er fordert vom Kerberos-Server ein sog. Ticket Granting Ticket (TGT) an. Hierzu muss der Nutzer des Clients entweder ein Passwort eingeben oder das TGT wird direkt bei der Benutzeranmeldung angefordert. Die Unterstützung von digitalen Zertifikaten auf Smartcards ist im UNIX-Bereich zurzeit in Entwicklung, wird aber bereits von Windows 2000 und 2003 unterstützt, und stellt damit die Hauptanwendung von Unternehmens-PKIs¹² dar. Im Juni 2006 wurde das Dokument von der IETF zum offiziellen Standard (RFC 4556) ernannt. Mit dem TGT ist der Client in der Lage, weitere Tickets für Dienste anzufordern, ohne nochmal ein Passwort eingeben zu müssen. Es wird auch ein Session Key für die Kommunikation zwischen Client und Kerberos-Server ausgehandelt. Er kann benutzt werden, um den Datenverkehr zu verschlüsseln.

Um einen Dienst, der Kerberos unterstützt, benutzen zu können, fordert der Client ein weiteres Ticket an. Dieses Ticket sendet der Client dann an den Dienst, der überprüft, ob er dem Client den Zugriff gestatten soll. Auch hierbei wird ein Sitzungsschlüssel vereinbart und die Identität von Client, Server und Kerberos-Server überprüft.

Erläuterungen zur Abbildung 2.8:

Szenario: Nutzer *u* möchte Service *s* nutzen, er besitzt noch kein TGT. Die kleineren Rechtecke (hellgrün, hell-orange, weiß) sind Datenpakete, die jeweils mit dem nach dem Stern (*) stehenden Schlüssel verschlüsselt sind. Das Kürzel ST meint hier: Ticket zur Nutzung des Services *s*. In den großen Rechtecken (Server) und in der Ellipse (Client) stehen nach den Pfeilen diejenigen Informationen, die dem jeweiligem Service/Client bekannt sind. Kerberos Authentication-Server und Ticket Granting Server (TGS) haben beide Zugriff auf die Schlüsseldatenbank ihres Administrationsbereiches (Realm), sie kennen also beide alle Client- und Server-Schlüssel.

Ein Kerberos-Server ist für einen Realm zuständig, d. h. er verwaltet nur Konten, die zu seinem Realm gehören. Der Realm kann beispielsweise der DNS-Domänen-Name in Großbuchstaben, etwa EXAMPLE.COM, sein. Ein Rechner kann immer nur zu einem Realm gehören. Um auf Dienste in anderen Realms über Kerberos zugreifen zu können, müssen Vertrauensstellungen zwischen den einzelnen Realms hergestellt werden. So ist es möglich, dass ein Benutzer aus A.EXAMPLE.COM auf Dienste in B.EXAMPLE.COM zugreifen kann, ohne sich erneut authentisieren zu müssen.

¹²PKI, engl. public key infrastructure

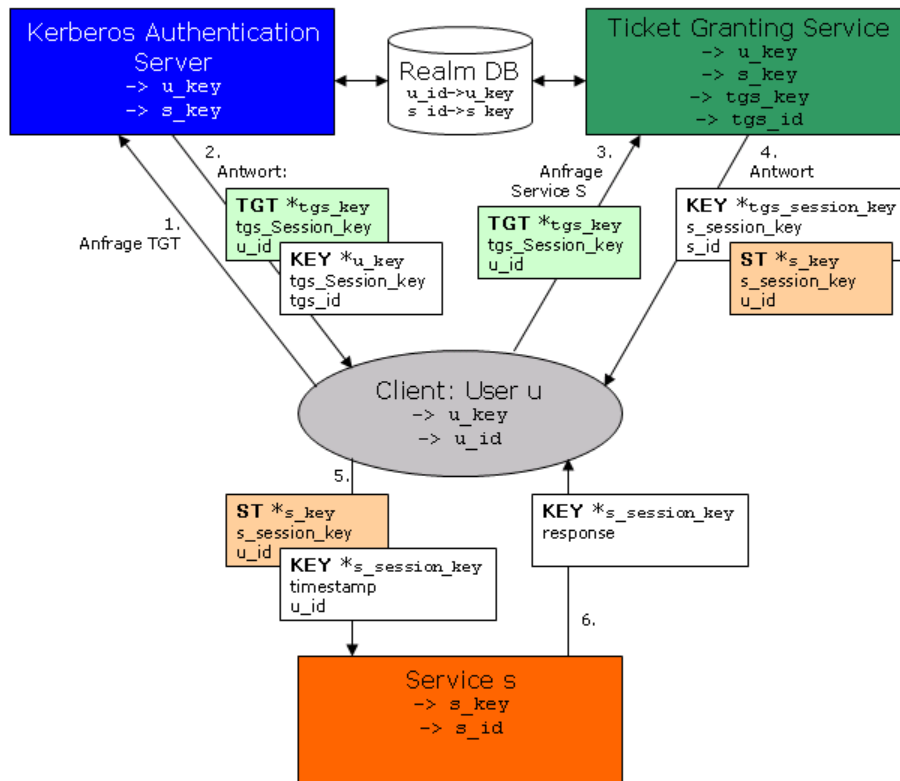


Abb. 2.8: Kerberos Schema

Bei Kerberos4 wird als Chiffre nur DES unterstützt. Kerberos5 ist in der Lage, die verwendete Chiffre und das verwendete Prüfsummenverfahren auszuhandeln.

Nutzer, Hosts und Dienste werden bei Kerberos über symmetrische Schlüssel authentifiziert. Dem Schlüssel ist ein Name, der Kerberos Principal, zugeordnet. Für Hosts ist der Principal `host/<hostname>@<REALM>` (z. B.: `host/www.example.com@EXAMPLE.COM`), für Dienste `<servicename>/<hostname>@<REALM>` (z. B.: `imap/www.example.com@EXAMPLE.COM`) und für Nutzer `<benutzer>/<instanz>@<REALM>` (z. B.: `mueller/admin@EXAMPLE.COM`). Die Instanz gibt bei einem Nutzer-Principal die Art des Accounts an. Der Nutzer `mueller/admin@EXAMPLE.COM` ist ein Kerberos-Administrator.

Durch Kerberos werden insbesondere Angriffe durch passives Sniffing unterbunden, aber auch Spoofing, Wörterbuch-, Replay- und andere Angriffe erschwert.

Weitere Details siehe *Kerberos Authentifikation*: <http://wwwbs.informatik.htw-dresden.de/svortrag/ai95/Bindrich/kerberos.html>.

[3], [4]

2.6 Die Registry

Die Windows-Registrierungsdatenbank (auch: Windows-Registry oder Windows-Registrierdatenbank) ist seit der ersten Version von Windows NT die zentrale hierarchische Konfigurationsdatenbank des Betriebssystems Microsoft Windows. Hier werden sowohl Informationen und Konfigurationsdaten von Windows selbst als auch Informationen und Konfigurationsdaten von Programmen gespeichert. Es handelt sich bei der Registry seit Windows 95 und Windows NT 4.0 um eine umfassende Datenbank für die Verwaltung des Systems und aller integrierten Systemdienste und -prozesse. Die Registry bietet auch die Möglichkeit, dort die Einstellungen der installierten Anwendungen zentral abzulegen.

2.6.1 Aufbau und Struktur

Registrierungs-Einträge werden in einer Baumstruktur in so genannten *Schlüsseln* (engl. keys) angelegt, die alle von einigen Hauptschlüsseln abstammen.

Die Registrierung besteht aus zwei Teilen: Der erste Teil umfasst Konfigurationsdaten für die gesamte Windows-Installation, der zweite Teil beinhaltet alle benutzerspezifischen Informationen und Einstellungen. Die in der Registrierungsdatenbank gespeicherten Daten enthalten alle variablen Informationen des Betriebssystems, wie zum Beispiel Größe und Name der Auslagerungsdatei, Einstellungen für den Windows-Explorer, die gesamte COM-Registrierung (Klassen und Typenbibliotheken), Einstellungen für diverse Programme, Treibereinstellungen und die Hardwarekonfiguration. Desweiteren ist die Registrierungsdatenbank hierarchisch aufgebaut. Beginnend in sogenannten Hauptschlüsseln, (Hives) verzweigt über Unterschlüssel. In der oberen Ebene befinden sich 5 Hive Keys. Die einzelnen *Hive Keys* sind:

HKEY_CLASSES_ROOT

Hier sind alle Informationen zu Dateiendungen gespeichert. Auch Treibereinstellungen und OLE Funktionen finden sich unter diesem Key. Diese Einstellungen legen z. B. fest, dass das richtige Programm gestartet wird, wenn man zum Beispiel eine Datei im Windows-Explorer öffnet.

HKEY_CURRENT_USER

In diesem Schlüssel sind alle Konfigurationsinformationen des gerade aktiven Benutzers gespeichert, genauer gesagt: Verweise auf die jeweiligen Einträge in HKEY_USERS. Hier werden die Ordner, die Bildschirmfarben und die Einstellungen der Systemsteuerung gespeichert. Diese Informationen werden auch ganz einfach als 'Profil' des Benutzers bezeichnet.

HKEY_LOCAL_MACHINE

Unter diesem Key werden die Systemeinstellungen des Rechners verwaltet, die für alle Benutzer gelten. Dazu gehört die Liste aller installierter Hardware und Treiber und die unterschiedlichen Hardwarekonfigurationen. Daneben findet man hier globale Softwareeinstellungen für Windows und andere Anwendungen.

HKEY_USERS

Hier sind alle Einstellungen sämtlicher Benutzerprofile gespeichert. Zu diesen Einstellungen

gehören unter anderem der Aufbau des Startmenüs und der Ordner, aber auch Farbeinstellungen des Desktops. Des Weiteren findet sich hier ein Unterschlüssel namens `.Default`, der die Standardeinstellungen für alle Benutzer enthält, die kein persönliches Profil erstellt haben.

HKEY_CURRENT_CONFIG

In diesem Schlüssel befinden sich Informationen zum Hardwareprofil, das vom lokalen Computersystem beim Start verwendet wird. Im Grunde sind es Verweise auf die gerade aktive Konfiguration in `HKEY_LOCAL_MACHINE`.

Diese Registry-Informationen setzen sich aus mehreren Dateien im Verzeichnis `Windows\System32\Config` zusammen. Siehe Tabelle 2.1 auf Seite 34.

Registrierungsstruktur	Zugehörige Datei(en)
<code>HKEY_LOCAL_MACHINE\SAM</code>	Sam, Sam.log, Sam.sav
<code>HKEY_LOCAL_MACHINE\Security</code>	Security, Security.log, Security.sav
<code>HKEY_LOCAL_MACHINE\Software</code>	Software, Software.log, Software.sav
<code>HKEY_LOCAL_MACHINE\System</code>	System, System.alt, System.log, System.sav
<code>HKEY_CURRENT_CONFIG</code>	System, System.alt, System.log, System.sav, Ntuser.dat, Ntuser.dat.log
<code>HKEY_USERS_DEFAULT</code>	Default, Default.log, Default.sav

Tab. 2.1: Zuordnung Registryhives und zugehörige Dateien

Daneben gibt es noch zwei weitere Datei, in der die Registryinformation des jeweiligen Anwenders abgelegt sind. Die `ntuser.dat` im Verzeichnis *Dokumente und Einstellungen* des jeweiligen Benutzerprofils und die Datei `usrclass.dat`, die Sie im jeweiligen Benutzerprofil unter *\Lokale Einstellungen\Anwendungsdaten\Microsoft\Windows* finden. Diese sieben Dateien werden in der Registry zu einer Art Datenbank zusammengefasst und für den Anwender übersichtlicher dargestellt.

[5]

2.6.2 Datentypen

Datentyp	Beschreibung
REG_BINARY	Rohe Binärdaten. Die meisten Informationen zu Hardwarekomponenten werden als Binärdaten gespeichert und im Registrierungseditor im Hexadezimalformat angezeigt.
REG_DWORD	Daten, die durch eine Zahl repräsentiert werden, die eine Länge von 4 Bytes hat (eine ganze Zahl mit 32 Bit). Viele Parameter für Gerätetreiber und Dienste weisen diesen Typ auf und werden im Registrierungseditor im binären, hexadezimalen oder dezimalen Format angezeigt.
REG_EXPAND_SZ	Eine Datenzeichenfolge variabler Länge. Dieser Datentyp beinhaltet Variablen, die aufgelöst werden, wenn ein Programm oder Dienst die Daten verwendet.
REG_MULTI_SZ	Eine multiple Zeichenfolge. Werte, die Listen oder mehrere Werte in einer Form enthalten, die für den Benutzer lesbar sind, weisen in der Regel diesen Typ auf. Die einzelnen Einträge werden dabei durch Leerstellen, Kommas oder andere Zeichen voneinander getrennt.
REG_SZ	Eine Textzeichenfolge festgelegter Länge.
REG_NONE	Daten ohne bestimmten Typ. Diese Daten werden durch das System oder Anwendungen in die Registrierung geschrieben und im Registrierungseditor im hexadezimalen Format als Binärwert angezeigt.

Tab. 2.2: Die wichtigsten Registry Datentypen

2.7 Benutzerverwaltung mit dem Active Directory

Windows steuert den Zugriff auf seine Ressourcen, Dateisystem, Registry, Drucker usw., mit Hilfe einer eigenständigen Sicherheitsdatenbank. Diese Sicherheitsdatenbank befindet sich entweder in der Registry auf Nicht-Domain Controllern oder in einer Jet (Joint Engine Technology) Datenbank auf Domain Controllern. Den Zugriff auf diese Datenbank steuert der Security Accounts Manager (SAM).

2.7.1 Lokale Nutzerverwaltung – Registry-basierte Nutzerverwaltung

Jeder Nicht-Domain Controller verwaltet Nutzer und Gruppen in der lokalen Nutzerverwaltung. Die dort angelegten Nutzer und Gruppen können nur für die Zugriffssteuerung auf lokale Ressourcen, wie Dateisysteme oder Drucker verwendet werden. Es ist nicht möglich lokale Nutzer oder Gruppen an andere Windows PCs im Netzwerk zu exportieren, um diese dann dort für die Zugriffssteuerung auf Ressourcen zu benutzen.

Den Export von Nutzern und Gruppen beherrschen nur Domain Controller, welche nur auf Windows Server Installationen eingerichtet werden können.

2.7.2 Active Directory – Datenbank-basierte Nutzerverwaltung

Der Verzeichnisdienst von Microsoft Windows 2000/Windows Server 2003 heißt Active Directory (AD). Ab der aktuellen Version Windows Server 2008 wird die Kernkomponente als Active Directory Domain Services (ADDS) bezeichnet. Active Directory ist eine datenbank-basierte Benutzerverwaltung.

Active Directory ermöglicht es, ein Netzwerk entsprechend der realen Struktur des Unternehmens oder seiner räumlichen Verteilung zu gliedern. Dazu verwaltet es verschiedene Objekte in einem Netzwerk wie beispielsweise Benutzer, Gruppen, Computer, Server, Dateifreigaben und andere Geräte wie Drucker und Scanner und deren Eigenschaften. Mit Hilfe von Active Directory kann ein Administrator die Informationen der Objekte organisieren, bereitstellen und überwachen.

Den Benutzern des Netzwerkes können Zugriffsbeschränkungen erteilt werden. So darf zum Beispiel nicht jeder Benutzer jede Datei ansehen oder jeden Drucker verwenden.

Aufbau

Bestandteile

Active Directory ist in drei Teile aufgegliedert: Schema, Konfiguration und Domäne. Ein Schema ist eine Schablone für alle Active-Directory-Einträge. Es definiert Objekttypen, ihre Klassen und Attribute als auch ihre Attributsyntax. Welche Objekttypen im Active Directory verfügbar sind, lässt sich durch die Definition neuer Typen beeinflussen. Das dafür zugrundeliegende Muster ist das „Schema“, das die Objekte und ihre Attribute definiert.

Die Konfiguration stellt die Struktur des Active-Directory-Waldes und seiner Bäume dar.

Die Domäne schließlich speichert alle Informationen über die erstellten Objekte und seiner Domäne.

Die ersten beiden Teile der Active Directory werden mit jedem Domänencontroller repliziert. Es gibt nur einen globalen Katalog, in dem alle Informationen der Domänen gespeichert werden. Die Grenze der vollen Domänenreplikation stellt die Domäne selbst dar.

Datenbank

Das Active Directory verwendet zur Speicherung der Informationen über die Netzwerkobjekte eine Jet (Blue)-Datenbank, die Microsoft auch für den Exchange Server einsetzt. Sie ist relational, transaktionsorientiert und benutzt ein „Write-Ahead-Logging“. Die Active-Directory-Datenbank ist auf 17 Terabytes und 10 Millionen Objekte pro Domäne begrenzt. Dies ist ein theoretischer Grenzwert, da nicht mehr als eine Million Objekte pro Domäne empfohlen werden.

Die Datenbankdatei *ntds.dit* enthält drei Haupttabellen: die „schema table“ zur Speicherung der Schemata, die „link table“ zur Speicherung der Objekt-Struktur und die „data table“ zur Speicherung der Daten.

ESE (extensible storage engine) ordnet die nach einem relationalen Modell abgespeicherten Active-Directory-Daten nach einem vorgegebenen Schema in einem hierarchischen Modell an.

Unter Windows 2000 benutzt Active Directory die Jet-basierende ESE98-Datenbank.

Objekte

Im Gegensatz zum objektorientierten Verzeichnissystem eDirectory von Novell, ist das Active Directory eher als objektbasiert – und hierarchisch – zu bezeichnen.

Die Datensätze in der Datenbank werden im Active Directory als „Objekte“ und deren Eigenschaften als „Attribute“ definiert. Die Attribute sind abhängig von ihrem Typ definiert. Objekte werden eindeutig über ihren Namen identifiziert.

Die Gruppenrichtlinien-Einstellungen werden in Gruppenrichtlinien-Objekten gespeichert. Diese sind ebenfalls Domänen und Standorten zugeordnet.

Objektkategorien

Objekte lassen sich in drei Kategorien einteilen:

- Ressourcen, wie zum Beispiel Computer, Server, Drucker, Scanner und Kameras
- Dienste, wie zum Beispiel E-Mail
- Konten, wie zum Beispiel Benutzerkonten, Gruppenkonten und Computerkonten

Ablage in Containern (Organisationseinheiten)

Die möglicherweise bis zu vielen Millionen Objekte werden in Containern (Organisationseinheiten), auch OUs (Organizational Unit) genannt, abgelegt. Einige Container sind vordefiniert, beliebige weitere Organisationseinheiten können mit Subeinheiten (Unterorganisationseinheiten) erstellt werden. Als objektbasiertes System unterstützt Active Directory die Vererbung von Eigenschaften eines Objektcontainers an untergeordnete Objekte, die auch wieder Container sein können. Dadurch erlaubt es Active Directory, Netzwerke logisch und hierarchisch aufzubauen.

Wald (forest)

Die gesamte hierarchische Struktur heißt „Wald“ (forest) oder auch Gesamtstruktur; eine Ansammlung aller Objekte, deren Attribute, Regeln und Container in dem Verzeichnis abgelegt werden. Der Wald verwaltet einen oder mehrere transitiv verknüpfte Bäume. Ein Baum verwaltet eine oder mehrere Domänen, welche wiederum transitiv in der Hierarchie miteinander verknüpft sind. Domänen werden nach den Regeln des DNS-Systems benannt, dem „Namensraum“ (Namespace).

Organisationseinheiten

Eine Organisationseinheit (OU) ist ein Containerobjekt, das zum Gruppieren anderer Objekte im AD dient. Eine OU kann neben Objekten auch andere OUs enthalten. Die frei definierbare Hierarchie der OUs vereinfacht die Administration von Active Directory. In der Regel richtet sie sich nach den Netzwerkstrukturen (Netzwerkverwaltungsmodell) oder nach der Organisationsstruktur des Unternehmens. Die OUs sind die unterste Ebene von Active Directory, in der administrative Rechte aufgeteilt werden können.

Domänencontroller und Replikation

Windows-NT

Unter Windows-NT gab es pro Domäne immer einen ausgezeichneten Controller, den primären Domänencontroller (PDC), der Änderungen an der Nutzer- und Computerdatenbank (SAM) ausführen durfte. Alle anderen Domänencontroller dienten als Sicherungskopie, die im Bedarfsfall zu einem PDC hochgestuft werden können.

Ab Windows 2000: Multimaster-Replikation

Active Directory nutzt für die Replikation des Verzeichnisses zwischen den Domänencontrollern eine sogenannte Multimaster-Replikation. Das hat den Vorteil, dass sich jedes Replikat beschreiben und synchronisieren lässt. Somit ist bei verteilten Implementierungen eine lokale Administration vollständig möglich. Im Gegensatz zu NT4-Domänen besitzen ab Windows 2000 alle Domänencontroller (DC) eine beschreibbare Kopie der Active-Directory-Datenbank. Die Veränderung eines Attributes auf einem der DCs wird in regelmäßigen Intervallen an alle anderen DCs weitergegeben (repliziert). Dadurch sind alle DCs auf demselben Stand. Der Ausfall eines DCs ist für die Active Directory Datenbank unerheblich, da keine Informationen verloren gehen. Das Replikationsintervall kann je nach Änderungshäufigkeit auf 15 oder mehr Minuten eingestellt werden. Windows 2000 Server repliziert das AD standardmäßig nach spätestens 5 Minuten, Windows Server 2003 repliziert es standardmäßig nach spätestens 15 Sekunden. Da eine Replikation über höchstens 3 Hops geht, erhält man je nach verwendeter Serverversion 15 Minuten bzw. 45 Sekunden als Replikationsintervall für eine Domäne.

Namensvergabe

Active Directory unterstützt eine Benennung und den Zugriff über UNC/URL- und LDAP-URL-Namen¹³. Intern wird die LDAP-Version X.500 für die Namensstruktur verwendet. Jedes Objekt hat einen vollqualifizierten Namen (distinguished name, DN). Ein Druckobjekt heißt beispielsweise „LaserDrucker3“ in der organisatorischen Einheit „Marketing“ und der Domäne „foo.org“. Der voll qualifizierte Name ist somit „CN=LaserDrucker3,OU=Marketing,DC=foo,DC=org“. „CN“ steht hierbei für *common name*. „DC“ ist die Domänen-Objekt-Klasse (*domain component*), die aus sehr vielen Teilen bestehen kann. Die Objekte können auch nach der UNC/URL-Notation bezeichnet werden. Diese zeichnet sich durch eine umgekehrte Reihenfolge der Bezeichner aus, welche durch Schrägstriche voneinander getrennt sind. Das obige Objekt könnte somit auch mit „foo.org/Marketing/LaserDrucker3“ bezeichnet werden. Um Objekte innerhalb der Container anzusprechen, werden relative Namen (*relative distinguished names*, RDNs) verwendet. Dies wäre für den Laserdrucker „CN=LaserDrucker3“. Jedes Objekt hat neben seinem global eindeutigen Namen eine ebenfalls global eindeutige 128 Bit lange Nummer (globally unique identifier, GUID). Diese wird üblicherweise als Zeichenfolge dargestellt und ändert sich auch beim Umbenennen des Objekts nicht. Weiterhin kann jedes Benutzer- und Computerobjekt auch eindeutig über seinen zugeordneten UPN (User Principal Name) angesprochen werden, der den Aufbau *Objektname@Domänenname* hat.

¹³LDAP: Das Lightweight Directory Access Protocol erlaubt die Abfrage und die Modifikation von Informationen eines Verzeichnisdienstes (eine im Netzwerk verteilte hierarchische Datenbank) über das TCP/IP-Netzwerk.

2.7.3 Zugriffsberechtigungen unter Windows – Das AGDLP Prinzip

Zugriffsberechtigungen werden am besten nur für Gruppen vergeben. Unter Windows sind in einer Domäne dabei einerseits zwei Gruppen von Gruppentypen interessant, andererseits die Zugriffsberechtigungen und Geltungsbereiche der Gruppen.

Es gibt unter Windows im Active Directory zwei Arten von Gruppen:

- Globale Gruppen
- Domänenlokale Gruppen

Der Unterschied in der Wirkungsweise der Gruppen liegt darin, daß globale Gruppen über Domänengrenzen hinweg wirken. Domänenlokale Gruppen hingegen wirken nur in der lokalen Domäne in der sie angelegt sind. Aufgrund dieser Tatsache wird von Microsoft das sogenannte AGDLP Prinzip beim vergeben der Berechtigungen unter Windows empfohlen.

Es werden Domänenlokale und Globale Gruppen angelegt. Den domänenlokalen Gruppen werden Berechtigungen per ACL zugeordnet. Weiter werden den domänenlokalen Gruppen die entsprechenden globalen Gruppen zugeordnet. Nur den globalen Gruppen werden Benutzerkonten zugeordnet. Der Sinn mit zwei Gruppentypen zu arbeiten besteht im Geltungsbereich der Gruppen. Globale Gruppen gelten im gesamten Bereich der Domäne. Domänenlokale Gruppe gelten nur in der lokal Vorort vorhandenen Domäne. Damit ist es möglich Berechtigungen sozusagen on-the-fly zu vergeben. Wenn die lokal in einer Domäne vorhandenen Berechtigungen für domänenlokale Gruppen entsprechend definiert sind, dann reicht schon ein hinzufügen einer domänenlokalen Gruppe zu einer globalen Gruppe im Active Directory um bestimmten Benutzern entsprechende Berechtigungen zu geben. Es müssen keine NTFS Rechte geändert werden. Theoretisch könnte man nun sagen, dass die domänenlokalen Gruppen unnütz sind, da man Berechtigungen auch allein auf globale Gruppen ausrichten kann. Allerdings müsste man dann mehrere globale Gruppen anlegen, und zweitens müssten dann alle globalen Gruppen im Global Catalog repliziert werden. Das würde nur unnötig Traffic verursachen vor dem Hintergrund, dass die Auswertung der entsprechenden ACLs nur im lokal vorhanden Domänencontroller stattfindet. Es ist also nicht nötig in dem Augenblick mit globalen Gruppen zu arbeiten. Es reicht ein Verschachteln der globalen mit der lokalen Gruppe. Ein weiterer Vorteil ist eine entsprechende Dokumentation im Active Directory.

[10]

Administrative und nicht-administrative Gruppen

Die Benutzerverwaltung unter Windows kann grob in privilegierte (administrative) Gruppen und nicht-privilegierte (nicht-administrative) Gruppen eingeteilt werden.

Die wichtigsten Gruppen unter Windows sind in Tabelle 2.3 auf Seite 41 zusammengefasst.

Administrative Gruppen sind alle Gruppen, deren Mitglieder globale Änderungen am System vornehmen können, dazu zählen:

- Administratoren
- Domänen-Administratoren
- Hauptbenutzer
- Netzwerkkonfigurations-Operatoren
- Sicherungs-Operatoren

Zu nicht-administrative Gruppen gehören alle Gruppen, deren Mitglieder keine globalen Änderungen am System vornehmen können. Dies sind u. a.

- Benutzer
- Gäste

Benutzerkonten erhalten ihre Privilegien über die Mitgliedschaft in Gruppen. Dies bedeutet: Jedes Mitglied der Gruppe *Administratoren*, erhält über Mitgliedschaft in der Gruppe *Administratoren* automatisch Vollzugriff auf das System.

[7], [8], [8]

Gruppe	Privilegien	Mitglieder
Administratoren	Mitglieder dieser Gruppe verfügen über Vollzugriff und können Benutzern nach Bedarf Benutzerrechte und Zugriffssteuerungsberechtigungen zuweisen.	Administrator
Benutzer	Mitglieder dieser Gruppe können die meisten allgemeinen Aufgaben durchführen, wie z. B. das Ausführen von Anwendungen, das Verwenden von lokalen und Netzwerkdruckern sowie das Sperren des PCs. Benutzer dürfen keine Verzeichnisse freigeben oder lokale Drucker erstellen.	Authentifizierte Benutzer
Gäste	Für Mitglieder dieser Gruppe wird bei der Anmeldung ein temporäres Profil erstellt, das gelöscht wird, wenn das Mitglied sich abmeldet. Das Gastkonto (standardmäßig deaktiviert) ist ebenfalls Standardmitglied dieser Gruppe.	Gast
Hauptbenutzer	Mitglieder dieser Gruppe können Benutzerkonten erstellen und anschließend die von ihnen erstellten Konten ändern und löschen. Sie können lokale Gruppen erstellen und anschließend den von ihnen erstellten lokalen Gruppen Benutzer hinzufügen oder aus ihnen entfernen. Sie können darüber hinaus den Gruppen Hauptbenutzer, Benutzer und Gäste Benutzer hinzufügen oder aus ihnen entfernen. Mitglieder können freigegebene Ressourcen erstellen und die von ihnen erstellten freigegebenen Ressourcen verwalten. Sie können nicht den Besitz von Dateien übernehmen, Verzeichnisse sichern oder wiederherstellen, Gerätetreiber laden oder entladen oder Sicherheits- und Überwachungsprotokolle verwalten.	keine

Tab. 2.3: Benutzergruppen unter Windows

Literaturverzeichnis

- [1] Wikipedia: *Network Address Translation*.
http://de.wikipedia.org/wiki/Network_Address_Translation, Stand: 2008-08-22
- [2] Wikipedia: *Proxy (Rechnernetz)*.
http://de.wikipedia.org/wiki/Proxy_%28Rechnernetz%29, Stand: 2008-02-15
- [3] Alexander Kolbe - HTW-Dresden: *Betriebssysteme 3 - Architektur von Windows XP*.
https://www.bs.informatik.htw-dresden.de/svortrag/ai99/Kolbe/betriebssysteme_3.htm, Stand: 2008-11-19
- [4] Wikipedia: *Kerberos*.
[http://de.wikipedia.org/wiki/Kerberos_\(Informatik\)](http://de.wikipedia.org/wiki/Kerberos_(Informatik)), Stand: 2008-11-19
- [5] Wikipedia: *Windows-Registrierungsdatenbank*.
http://de.wikipedia.org/wiki/Windows_Registry, Stand: 2009-01-02
- [6] Brown, Keith: *Programming Windows Security*. Addison Wesley, 2. Auflage, November 2000, ISBN 0201604426
- [7] Wikipedia: *Active Directory*.
http://de.wikipedia.org/wiki/Active_Directory, Stand: 2009-01-12
- [8] Wikipedia: *Lightweight Directory Access Protocol*.
<http://de.wikipedia.org/wiki/LDAP>, Stand: 2009-01-12
- [8] Uni Rostock: *Benutzerverwaltung - Administrative und nicht-administrative Benutzer*.
<http://www.uni-rostock.de/Rechenzentrum/sware/WindowsNT/Security/UserManagement.shtml>, Stand: 2009-01-12
- [9] Richter, Jeffrey: *Microsoft Windows Programmierung für Experten*. Microsoft Press, Redmont Washington, 2000 (4. vollständig überarbeitete Ausgabe), ISBN 3-86063-615-4
- [10] w3-creative: *AGDLP Prinzip*.
<http://www.w3-creative.de/agdlp/>, Stand: 2009-03-07

Stichwortverzeichnis

- Active Directory, 30, 35, 36
 - AGDLP *siehe* Zugriffsberechtigungen 39
 - Aufbau, 36
 - Benutzerverwaltung, 36
 - Bestandteile, 36
 - Datenbank, 36
 - Domänencontroller, 38
 - Domänlokale Gruppe, 39
 - forest, 37
 - Globale Gruppe, 39
 - Namensvergabe, 38
 - Objekte, 37
 - Objektkategorien, 37
 - Organisationseinheiten, 37
 - Primärer Domänencontroller, 38
 - Replikation, 38
 - Zugriffsberechtigungen, 39
- Active Directory Domain Services, 36
- AD *siehe* Active Directory 36
- ADDS *siehe* Active Directory Domain Services 36
- Adressumsetzung, 10
- APIC, 23
- Application Gateway, 10, 12–14, 17
 - Stärken, Schwächen, 15
- Arbeitsspeicher, 21
- Auslagerungsdatei, 23

- Betriebssystem, 21
- BIOS, 22
- boot.ini, 23
- Bootloader, 22
- Bootmanager, 23
- Bootmenü, 23

- CPU, 21

- DC *siehe* Domänencontroller 38
- demilitarisierte Zone, 16

- Filterregel, 13
- Firewall
 - Aufgabe, 7
 - Definition, 7
 - Funktion, 7
 - Komponenten, 12
 - Typen, 10

- Globally Unique Identifier, 38
- GUID *siehe* Globally Unique Identifier 38

- HAL *siehe* Hardware Abstraction Layer 26
- hall.dll, 23
- Hardware Abstraction Layer, 23, 26

- Infrastruktur, 9

- Jet *siehe* Joint Engine Technology 35
- Joint Engine Technology, 35

- Kerberos, 30
- Kernel
 - Microkernel, 27
 - monolithisch, 27
- Kernel-Mode, 23, 26
- Kommunikationsanforderungen, 8
- Konfigurationsdatenbank, *siehe* Registry

- LDAP *siehe* Lightweight Directory Access Protocol 38
- Lightweight Directory Access Protocol
 - common name (CN), 38
 - domain component (DC), 38
 - Relative distinguished name (RDN), 38
 - User Principal Name (UPN), 38
 - vollqualifizierter Name (DN), 38
- Local Procedure Call, 26
- Local Security Authority, 26
- Logon Session, 26
- LPC *siehe* Local Procedure Call 26
- LSA *siehe* Local Security Authority 26

- Mehrbenutzerbetriebssystem, 21
- Microsoft, 21
- Motherboard, 21
- MSGina.dll, 25
- Multi Homed Application Gateway, 19
- Multitasking, 21

- Netzwerk Topologien, 15
- ntdetect.com, 23
- NTDLR, 23
- ntds.dit, 37
- ntkrnl.exe, 23
- ntuser.dat, 34

- Organisation, 9

- Pagefile, 23
- Paketfilter, 10, 12, 13, 15
 - Eigenschaften, 13
 - Stärken, Schwächen, 14
- Partitionstabelle, 22
- PDC *siehe* Active Directory
 - Primärer Domänencontroller 38
- Personal, 9
- POST *siehe* Power on self test 22
- Power on self test, 22
- Proxy, 10, 14
 - Application Level Proxy, 14
 - Circuit LevelProxy, 14

- Real-Mode, 22
- Registrierungsdatenbank, *siehe* Registry
- Registry, 23, 33
 - Aufbau, 33
 - Datentypen, 35
 - Hauptschlüssel, 33
 - Hive, 33
 - Schlüssel, 33
 - Unterschlüssel, 33
- Remote Procedure Call, 26
- RPC *siehe* Remote Procedure Call 26

- SAM *siehe* Security Account Manager 35
- SAS *siehe* Secure Attention Sequence 25
- Screened Subnet, 18, 19
- Secure Attention Sequence, 25
- Security Account Manager, 35
- Sicherheitsanforderungen, 8

- Sicherheitsautorität, 24
- Sicherheitspolitik, 8
- Sicherheitsrichtlinien, 7
- Single Homed Application Gateway, 18
- Sitzungs-Manager, 23
- Speicher
 - physischer, 28
- Subsystem, 26
- Systempartition, 22

- User-Mode, 24, 26
- usrclass.dat, 34

- Verzeichnisdienst, 36

- Windows
 - 2000, 21
 - Adressraum
 - virtuell, 28
 - Arbeitsspeicher, 28, 29
 - Architektur, 26
 - Filesystem Manager, 26
 - Graphics Device Drivers, 27
 - I/O Manager, 26
 - IPC Manager, 26
 - Memory Manager, 26
 - Object Manager, 27
 - Plug and Play Manager, 27
 - Power Manager, 27
 - Security Reference Monitor, 27
 - Windows Manager, 27
 - Auslagerungsdatei, 28, 29
 - Authentifizierung, 25
 - Benutzerverwaltung, 35
 - Administrative- und nicht-administrative Gruppen, 39
 - Benutzerkonto, 40
 - Pagefile, 28
 - Seitenfehler, 28, 29
 - Sicherheitsdatenbank, 35
 - Speicherseite, 28, 29
 - Speicherverwaltung, 28
 - Startvorgang, 22
 - XP, 21
- Windows XP, 26
- winlogon.exe, 25