



Fakultät für Informatik
Professur Modellierung und Simulation

Diplomarbeit

Untersuchungen zur Integration der Active Directory
Technologie in das Dienstespektrum des
Universitätsrechenzentrums

Marko Damaschke

Chemnitz, den 16. Januar 2006

Prüfer: Prof. Dr. Peter Köchel
Betreuer: Dipl.-Ing. Christoph Ziegler

Damaschke, Marko

Untersuchungen zur Integration der Active Directory Technologie in das Dienstespektrum
des Universitätsrechenzentrums

Diplomarbeit, Fakultät für Informatik

Technische Universität Chemnitz, März 2006

Aufgabenstellung

Unter Berücksichtigung des vom Universitätsrechenzentrum (URZ) der TU Chemnitz angebotenen Dienstespektrums ist dessen Erweiterung durch Active Directory (AD) zu untersuchen. Dabei soll die vorhandene Infrastruktur nicht signifikant geändert werden. Für AD-basierende Technologien sind Alternativen zu diskutieren.

Folgende Teilaufgaben sind im Einzelnen zu realisieren:

- Untersuchung von AD-Nutzungsszenarien, die für die TU Chemnitz typisch sind
- Aufbau einer vom URZ als Dienst betriebenen AD-Zelle, in die sich bei Bedarf Struktureinheiten der TU Chemnitz in Form einer Subzelle oder Domäne integrieren können
- Authentifizierung auf Basis einer Windows-externen Kerberos5-Technologie
- Integration in eine existierende zentrale DNS-Technologie
- Anleitung zum Aufbau einer AD-Subzelle

Inhaltsverzeichnis

Abbildungsverzeichnis	v
1 Hintergründe zur Arbeit	1
1.1 Gedanken zur Aufgabenstellung	1
1.2 Ziel der Arbeit	1
1.3 Gründe für die Bereitstellung eines Active Directory Dienstes	2
2 Grundlagenbetrachtungen	4
2.1 Verzeichnisdienste	4
2.1.1 X.500	6
2.1.2 LDAP	8
2.2 Authentifizierung	10
2.2.1 Das Kerberos-Protokoll	10
2.2.2 Alternativen zu Kerberos	14
2.3 Authorisierung	15
3 Ausgangslage und Nutzungsszenarien im URZ	16
3.1 Einsatzszenarien	16
3.1.1 Einsatz von Windows-Servern in Struktureinheiten	16
3.1.2 Vertrauensstellung zwischen Struktureinheiten	17
3.1.3 Bereitstellung von AD-basierter Software	17
3.1.4 Active Directory als zentraler Verzeichnisdienst	17
3.2 Aktueller Stand im URZ	18
3.2.1 Authentifizierung	18
3.2.2 Authorization	18
3.2.3 Domain Name Service	19
3.2.4 Verzeichnisdienst	19
3.2.5 Softwareverteilung	19
4 Active Directory im Detail	21
4.1 Struktur einer Active Directory Domäne	23
4.2 Der globale Katalog	25
4.3 DNS und AD	27
4.4 Replikation, Betriebsmasterrollen und Backup	30
4.5 Funktionsebenen	32
4.6 Authentifizierung	33
4.7 Gruppenrichtlinie	33
4.8 Anmerkungen zu Hardwarevoraussetzung eines Domaincontrollers	35

5	Lösungsideen und Alternativen	37
5.1	Grundidee einer Gesamtlösung	37
5.2	Die Teststellung	37
5.3	Intuitiver Versuch	39
5.4	Namensraum - Vereinheitlichung der Namensgebung	39
5.5	DNS - dynamisch versus statisch	41
5.6	Kerberos im genannten Kontext	44
5.6.1	Windows und Schlüsseltypen	44
5.6.2	Windows mit externem Kerberos	45
5.6.3	Heimdal- versus MIT-Kerberos	47
5.6.4	Passwortverwaltung mit externem Kerberos	50
5.6.5	Anmerkung zu OpenAFS & Windows Server 2003	50
5.7	Benutzerverwaltung	51
5.8	Gruppenarten und Arbeiten mit Gruppen	55
5.9	Strukturierung mittels Organisationseinheiten (OU)	56
5.10	Firewall-Aspekte	58
5.11	AD als LDAP-Ersatz	59
5.12	Alternativen zu Active Directory	60
5.12.1	PADL/XAD	61
5.12.2	SAMBA	62
5.12.3	Keine Domäne	62
6	Fazit	64
6.1	Allgemeines	64
6.1.1	Anmerkungen zum Schema	65
6.1.2	Alternativen zu Active Directory	65
6.2	Strukturdebatte	66
6.2.1	Namensraum	66
6.2.2	Aufbau einer zentralen AD-Gesamtstruktur mit Subdomänen	67
6.2.3	Aufbau von individuellen Domänen/Gesamtstrukturen	69
6.2.4	Nutzung von Organisationseinheiten (OUs) für kleinere Struktureinheiten	70
6.3	Lösungsvorschlag zu einer Struktur	71
6.4	Aufbau einer zentralen Domäne	73
6.5	Domänen bei Struktureinheiten	76
6.5.1	Integration in einen Domänenbaum	77
6.5.2	Erstellen eigener Domänen	78
6.6	Einstellungen an den Clienten	79
6.7	Standorte	81
	Literaturverzeichnis	82
A	Bindkonfiguration der Teststellung	89
A.1	named.conf	89
A.2	db.spielwiese.netz	89
A.3	db.ad.spielwiese.netz	89

B Visual Basic Skripte	90
B.1 Das Skript zur Datenübernahme	90
B.2 Umgebende Text-Dateien	95
B.2.1 Nutzer.txt	95
B.2.2 lokaleNutzer.txt	95

Abbildungsverzeichnis

2.1	Aufbau eines Verzeichnisdienstes	5
2.2	Struktur eines X.500-Verzeichnisses	7
2.3	LDAP als Protokollkonverter	8
4.1	Eine Organisationseinheit in einer Domäne	24
4.2	Standort mit 2 Domänen	24
4.3	Domäne an 2 Standorten	25
4.4	Mehrere Standorte und Domänen	25
4.5	Eine Organisationseinheit in einer Domäne über mehrere Standorte erstreckt	26
4.6	Zwischenspeichern der universellen Gruppen aktivieren	27
5.1	Eine Möglichkeit einer Gesamtlösung	37
5.2	Struktur der Teststellung	38
5.3	Trennung von DNS-Suffix und Domainnamen	40
5.4	Anmeldung eines Kerberos-Nutzers an einem Windows-Clients	46
5.5	AD als Adressbuch eines Mailprogramms	60
6.1	Active Directory Baum mit Subdomänen	67
6.2	Parallele Active Directory Domänen	69
6.3	Organisationseinheit in einer Domäne	71
6.4	Einrichtung einer neuen Gesamtstruktur	74
6.5	Einstellungen zum globalen Katalog	75
6.6	Einrichtung der DNS-Weiterleitungen im MS-DNS-Server	77
6.7	Einbindung einer neuen Subdomäne	78
6.8	Entkopplung von DNS-Suffix und Domainnamen	80

1 Hintergründe zur Arbeit

1.1 Gedanken zur Aufgabenstellung

Das Universitätsrechenzentrum (URZ) der TU Chemnitz stellt als zentrale Einrichtung verschiedenste Dienste rund um das Thema der Rechentechnik bereit. Unter Anderem werden die administrative Betreuung von Pool- und Arbeitsplatzrechnern, die Betreuung und Verwaltung des CLiCs (Chemnitzer Linux Cluster) sowie verschiedenste und offensichtliche Netzdienste wie E-Mail-, Web- und Datenbankserver übernommen, aber auch die notwendigen Infrastrukturdienste wie eine zentrale Nutzerverwaltung, die Pflege verschiedenster Softwaresysteme, ein zentrales Authentifizierungssystem, ein verteiltes Filesystem, ein Zugangssystem sowie Netzinfrastrukturdienste wie DNS und DHCP.

Außerdem ist das Rechenzentrum Ansprechpartner für Fragen der Fachbereiche und unterstützt bei der Umsetzung individueller IT-Lösungen von Fakultäten, Fachgruppen oder Professuren.

Gerade im Bereich der Desktop-Arbeitsplätze aber auch der Anwendungsserver und Standardsoftwaresysteme ist der Einsatz von Windows XP und Windows Servern erwünscht und unumgänglich. Da deren Administration mit Windows XP und Windows 2000 konzeptionell stark auf den Einsatz des Active Directory Dienstes aufsetzt, bestimmte Software diesen gar voraussetzen, soll diese Arbeit die Integrationsmöglichkeiten der Active Directory Technologie in das Dienstspektrum des Universitätsrechenzentrums untersuchen.

Das Ziel der Arbeit soll dabei eine Erweiterung des Dienstangebots darstellen, ohne die vorhandene Infrastruktur signifikant zu ändern.

Es ist zu betrachten, was ein Verzeichnisdienst ist und wie die Active Directory Technologie in diese Klasse eingeordnet werden kann. Dann ist zu untersuchen, welche Nutzungsszenarien im URZ für den Einsatz eines Active Directory (AD) vorhanden sind und wie eine AD-Domäne in das Dienstspektrum integriert werden kann. Diese soll so ausgerichtet sein, dass bei Bedarf Struktureinheiten der TU Chemnitz sich mittels Subdomänen anschließen können, die Authentifizierung auch in der AD-Domäne und deren Subdomänen in die zentrale Heimdal Kerberos V5 Technologie integriert werden kann und eine Integration in die existierende zentrale DNS-Technologie stattfindet.

Abschließend soll eine Anleitung zum Aufbau und Integration einer Subdomäne erstellt werden, die den Systemadministratoren in den Struktureinheiten als Handreichung dienen soll.

1.2 Ziel der Arbeit

Wie aus dem Titel bereits abzulesen, ist das Ziel, die Integration der Active Directory Technologie in das Dienstespektrum des Universitätsrechenzentrums (URZ) der TU Chemnitz

zu untersuchen. Dabei soll die technische Realisierbarkeit betrachtet und ein Vorschlag zur Realisierung gemacht werden, unter der Maßgabe, die vorhandene Dienststruktur nicht wesentlich zu ändern, sondern zu erweitern.

Der Active Directory Dienst stellt einen Infrastrukturdienst dar, also ein Angebot, welches die Bereitstellung weiterer Dienste ermöglichen soll. Der Wunsch nach einem Active Directory besteht nicht nur seitens des URZ, sondern wurde auch bereits mehrfach von Struktureinheiten angetragen.

1.3 Gründe für die Bereitstellung eines Active Directory Dienstes

Mit der Einführung eines Active Directorys soll ein zentraler Verzeichnisdienst bereitgestellt werden, der für verschiedene Softwareprodukte notwendig ist beziehungsweise den Betrieb anderer vereinfachen soll. Beispielsweise nutzen viele E-Mail-Produkte ein zentrales Verzeichnis, um verschiedenste Informationen zu erlangen. So können User-Agenten eine Namenssuche via Verzeichnisdienst anbieten, Mailserver können Adressen auf solcher Basis umschreiben oder Mailinglisten verwalten. Eine andere große Gruppe sind die Groupware-Lösungen, von denen das URZ in der Folge der Arbeit auch eine zentrale Bereitstellung evaluiert.

Die Wahl speziell des Active Directorys als ein solcher Dienst basiert auf der Tatsache, dass verschiedene Produkte des Herstellers Microsoft sich auf diesen Dienst abstützen. So ist es nur mit einem Active Directory möglich, das Groupware-Produkt „Exchange“ zu nutzen. Daraus entstand seitens mehrerer Struktureinheiten der TU Chemnitz der Wunsch zu einer zentralen Bereitstellung eines solchen Dienstes. Weiterhin wurden mit Einführung von Windows XP und Windows 2000 viele konzeptionelle Ansätze zur Administration von einem solchen Verzeichnis abhängig gemacht. Jeder Domaincontroller mit Win2000-Server und nachfolgend erfordert für seinen Betrieb ein Active Directory. Eben auch die Nutzerauthentifizierung im Windows gegen eine Domäne wird auf ein Verzeichnis abgebildet, genauso bietet sich dadurch die integrative Möglichkeit der Softwareverwaltung auf Windows-Rechnern. Der absehbare Vorteil eines Active Directorys besteht in der vollkommenen Windows-Integration, die eben genannte Vorteile beim Einsatz von Windows-Arbeitsplatz-Rechnern aber auch Servertechnologien mit sich bringt und auf Grund dieser Integration auch ohne eigene Anpassungen mit Microsoft-Patches für deren Software-Produkte zusammenarbeitet.

Neben den genannten Vorteilen für die Windows-Systemwelt ergibt sich wegen der Wahl des LDAP-Protokolls als offene Schnittstelle des Active Directorys auch die Möglichkeit eines Nutzens beim Einsatz von Produkten auf anderen Plattformen. So greifen auch Groupware-Lösungen, die auf Nicht-Windows-Systemen laufen, auf Verzeichnisse mittels LDAP zurück. Als Vertreter seien hier die Open-Source-Entwicklungen „E-Groupware“ und „Open X-Change“ genannt.

Dieser Aspekt, dass die Bereitstellung eines Active Directorys auch einen Nutzen für Nicht-Windows-Systeme bietet, zeigt einen begrüßenswerten Trend bei der Konzeptionierung von Microsoft-Produkten auf. So wurde der Zugriff auf das AD mittels LDAP standardisiert. Weitere Dienste arbeiten nach offenen Standards oder wurden an diese angelehnt. Wie je-

de Implementierung eines Standards haben auch die Microsoft-Lösungen ihre Eigenarten und Anpassungen an eigene Gegebenheiten, zeigen aber, dass man zunehmend die Integration von Fremdprodukten bereits bei der Konzeptionierung bedenkt. Dies ist sicher auch in dem Druck zu erklären, dass nicht alle Umgebungen vollständig durch MS-Produkte bereitgestellt werden können und bei der Planung von IT-Landschaften eben auch „Altlasten“ mit einzuplanen sind. Durch diese Öffnung und Standardisierung der Schnittstellen wird es im Umkehrschluss auch attraktiver MS-Lösungen in gemischten Umgebungen als zentrale Dienste einzusetzen.

Die Vorteile, die ein zentrales Verzeichnis einer Organisation bringt, sind im Bereich der Infrastruktur ziemlich wesentlich. Wie jede Lösung haben Verzeichnisse ihre Vor- und Nachteile, doch kann bei einer Entscheidung für ein zentrales Verzeichnis der Aufwand für verschiedene Administrationsaufgaben verringert werden.

Verzeichnisse können einen Kernbestandteil eines zentralen Identity-Management darstellen. Mit Hilfe dieses Management kann eine höhere Aktualität und eine Reduzierung von Redundanzen von Daten und daraus resultierend ein höherer Nutzen für viele Anwendungen erreicht werden. So kann ein Verzeichnis, somit eben auch ein Active Directory, Teil einer zentralen Authentifizierung, dadurch auch einer Single-Sign-On-Lösung, sein. Mit einer zentralen Identity-Verwaltung können Verwaltungsaufgaben von verschiedensten Teilstrukturen auf eine zentrale Stelle verlagert, damit Kosten und Aufwände reduziert werden. Auch lassen sich dadurch Ressourcen nicht nur zentral verwalten, auch dezentral verwaltete Ressourcen können strukturübergreifend verfügbar gemacht und genutzt werden. Dazu können dann beispielsweise vorhandene Nutzerdaten zur Rechtevergabe als Grundlage herangezogen werden. Auch die Zusammenarbeit mehrerer Struktureinheiten lässt sich erleichtern, wenn deren Ressourcen gegenseitig zur Nutzung verfügbar gemacht werden können oder Anbieter unterschiedlicher Dienste über so genannte Trusts sich in ihrem Angebot vervollständigen können.

Visionär ist sogar damit zu rechnen, dass Identity-Managements auf sehr hoher zentraler Ebene angesiedelt werden, deren Datenbereitstellung dann in unterschiedlichsten Organisationen genutzt werden. Hier sei der Gedanke der elektronischen Bürgererfassung nur ein wenig weitergedacht. Ist dann in den Organisationen bereits das Identity-Management zentralisiert, kann der Aufwand einer Einbindung in ein noch höher angesiedeltes Identity-Management vereinfacht werden und transparenter geschehen.

Für einen weiteren Überblick der Einsatzszenarien, die ebenfalls als Gründe des Einsatzes sprechen, sei hier auf den Abschnitt 3.1 verwiesen.

2 Grundlagenbetrachtungen

Der zentrale Dienst eines Active Directorys ist ein Verzeichnisdienst. Allerdings wird diese zentrale Aufgabe von vielen Diensten begleitet, die in einer Wechselbeziehung mit dem Verzeichnis stehen, also entweder auf den Verzeichnisdienst aufbauen beziehungsweise dessen Einsatz ermöglichen. Manche dieser Beziehungen sind sogar im Windows Server Konzept wechselseitig.

In diesem Kapitel sollen Begriffe eingeführt, Protokolle genannt und Konzepte vorgestellt werden, die notwendig sind, um Active Directory und die vorhandene Struktur zu verstehen und die Stellen aufzuzeigen, an denen eine Integration möglich wäre. Es zeigt auch Hintergründe auf, die den Einsatz von Active Directory erklären.

Dieses Kapitel dient der Schaffung einer einheitlichen Begriffsbasis und der theoretischen Vorüberlegung.

2.1 Verzeichnisdienste

Wie bereits gesagt, ist die zentrale Dienstkomponente des AD ein Verzeichnisdienst. Somit erscheint es sinnvoll, hier eine Erläuterung des Begriffes Verzeichnisdienst, Verzeichnis anzureißen und auf die beiden hauptsächlich vertretenen Standards einzugehen.

So geht [Gar03] wie folgt kurz darauf ein: „Directories contain data describing resources, such as computers, printer and user accounts that are contained within a particular network.“ Dieses zeigt bereits auf den Sinn eines Verzeichnisdienstes innerhalb einer Organisation, die Bereitstellung von Informationen in einer klaren Struktur.

Zur Klärung des Begriffes Verzeichnisdienst sollte klar sein, was ein Verzeichnis ist. Jeder Einzelne hat sicher häufig mit Verzeichnissen zu tun. Es gibt sie auch nicht nur im elektronischen Umfeld, denken wir an Telefonbücher, Fernsehzeitungen oder Gebäudewegweiser, die einer Person ein Raum zuordnen.

Aber auch im elektronischen Umfeld hat man an verschiedenen Stellen mit Verzeichnissen zu tun - im Betriebssystem bei der Dateiablage, in Datenbanken bei der Datenspeicherung. Jeweils, wenn es um die strukturierte Ablage von Daten geht, um ein schnellstmögliches Wiederfinden nach bestimmten Suchkriterien zu gewährleisten, wird auf ein Verzeichnis zurückgegriffen.

Neben dem Verzeichnis als Datenablage, geht es hier um den Begriff des Verzeichnisdienstes. Wie [MR99] auf Seite 18 erwähnt, bestehen Verzeichnisdienste aus 2 Komponenten, dem Verzeichnis und dem Zugriffsdienst. Zum besseren Verständnis ist dies in Abbildung 2.1 nochmal dargestellt. Das Verzeichnis ist dabei der Datenbehälter, welcher in verschiedener Implementierung vorliegen kann. So sind Textdateien, binäre Dateiformate aber auch ein Datenbank-Backend als Verzeichnis denkbar. Der Zugriffsdienst wiederum definiert

das Protokoll, wie und wer auf das Verzeichnis zugreifen kann, sowie, welche Manipulationen an den Daten zulässig sind. Grundsätzlich ist festzuhalten, dass ein Verzeichnisdienst sich von einer Datenbank durch häufigere Lese- statt Schreiboperationen unterscheidet.

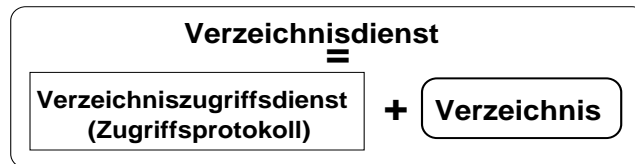


Abbildung 2.1: Aufbau eines Verzeichnisdienstes

Verzeichnisse sollen die verschiedenen Datenquellen einer Organisation zusammenführen, damit Redundanzen, Inkonsistenzen vermeiden und die Datenaktualität erhöhen. Als weitere Vorteile nennt [MR99] das schnelle und sichere Auffinden einer Information, die Möglichkeiten der mehrdimensionalen aber auch unscharfen Suche, den offenen Zugriff, also die Erreichbarkeit der Information, deren Replikation, die Verfügbarkeit verschiedener Informationsformen, also die Möglichkeit unterschiedliche Medien zu unterstützen wie Texte, Bilder, Videos aber auch andere Binärformate. Auch die Kostenersparnis, geringe Wartungskosten und kostengünstige Auskunft werden genannt, wobei letzteres darauf abzielt, dass nicht der Einkauf des gesamten Verzeichnisses für den Erhalt einer Einzelinformation notwendig ist. Die Liste der Vorteile wird ergänzt durch die Informationsintegration. Dabei wird ein Aspekt von Verzeichnisdiensten angesprochen, der bisher nicht genannt ist. So gibt es anwendungsspezifische Verzeichnisdienste, die auch allgemein bekannt, aber vielleicht nicht als solche gesehen werden. Ein typisches Beispiel ist der Domain Name Service (kurz DNS) oder das Filesystem. Die zweite Form sind die offenen Verzeichnisse, die keine spezielle Vorgabe hinsichtlich der darin gespeicherten Informationen treffen. Vielmehr folgen solche Verzeichnisdienste dem objektorientierten Ansatz, beschreiben Strukturen durch die Anordnung von Objekten und deren Querverknüpfung. Die Anordnung, die Form der Objekte und deren Ausgestaltung sind nicht vorgegeben, werden vom Verzeichnisadministrator definiert und im Verzeichnis-Schema abgelegt. Außerdem bieten die offenen Verzeichnisdienste die Möglichkeit der Integration von Informationen, die wiederum von anwendungsspezifischen Verzeichnissen genutzt werden. So kann ein offener Verzeichnisdienst als Master-Verzeichnis dienen, also die Informationen strukturiert auch für anwendungsspezifische Verzeichnisse bereitstellen und somit als „Single Point for Administration“ dienen.

Genau aus diesem Grund kann ein Verzeichnisdienst einen sehr zentralen Punkt in einer IT-Infrastruktur einnehmen. So lassen sich verschiedenste Informationen in einem solchen Master-Directory an einer Stelle administrieren. Die Vielfalt geht von Nutzerdaten, die zur Information dienen, über Authentifizierungs- und Autorisierungsdaten bis hin zu Informationen über Ressourcen, wie Drucker, Rechner, Datenspeicher et cetera. Beispielsweise kann eben auch der DHCP-Server oder DNS-Server mit Informationen aus einem Master-Verzeichnis versorgt werden, genauso wie ein Mailserver (MTA), der beispielsweise Adressen umschreiben oder Mailinglisten bedienen muss.

Weiterhin besteht bei einem Verzeichnisdienst die Möglichkeit, administrative Aufgaben zu

delegieren, was ein Argument zur Kostenersparnis ist. Jeder Nutzer kann in die Lage versetzt werden, seine eigenen personengebundenen Daten zu pflegen. Außerdem ist es beispielsweise denkbar, Mitarbeitern, die ihren eigenen Arbeitsplatz-PC administrieren, schreibenden Zugriff auf bestimmte Attribute dieses Rechners im Verzeichnis zu gewähren, um damit eine größere Aktualität zu erreichen.

Fällt die Entscheidung zu Gunsten eines derart zentralen Verzeichnisses, hat dies weitreichende Auswirkungen, die wohl bedacht und in allen Phasen der Einführung und der ersten Betriebszeit gut begleitet und vermittelt werden wollen.

Da mit der Einführung von Active Directory in Windows 2000 ein derart zentrales Verzeichnis eingebracht wurde, ist dessen Einsatz genau zu planen. Vertiefend wird in Kapitel 4 darauf eingegangen.

In den folgenden Absätzen geht es um die beiden häufigst vertretenen Standards bei offenen Verzeichnissen: Dem X.500 mit seinem Zugriffsprotokoll DAP sowie dem einst als „abgespeckte“ Variante abgeleiteten LDAP.

2.1.1 X.500

X.500 ist ein Standard der ISO und ITU-T, der den Entwurf eines globalen Verzeichnisdienstes beschreibt. Es handelt sich dabei um keinerlei technische Implementierung, sondern um Gestaltungsrahmen für die Konzeption eines Verzeichnisdienstes. X.500 ist offengelegt und vielfältig implementiert worden; viele Hersteller haben im Bereich der Administration ihrer Infrastruktur eine eigene Implementierung im Angebot.

Um dem offenen Ansatz gerecht zu werden, gibt es keine festen Vorgaben hinsichtlich der zu speichernden Informationen. Vielmehr gibt es allgemein nur Objekte und Verbindungen zwischen diesen. Ein Objekt ist dabei der Informationsträger, also etwas, worüber Informationen im Verzeichnis abgelegt werden. Jedes Objekt kann einer oder mehreren Objektklassen angehören, vergleichbar mit Masken, die beinhalten, welche Informationen über ein Objekt abgelegt werden können. Die einzelne Information zu einem Objekt entspricht einem Attributwert. Ein Attributwert hat einen bestimmten Datentyp und wird über einen Attributnamen referenziert. Die Summe aller Attribute, die im Objekt definiert sind, beschreiben dieses. Dabei gibt es notwendige und optionale Attribute.

Betrachtet man es am Beispiel einer Person, könnten etwa der Vor- und Nachname, das Geburtsdatum und eine Anschrift notwendige Attribute sein, dagegen die Augenfarbe, Körpergröße oder Telefonnummer optionale.

Wenn man alle Objekte eines Verzeichnisses als Gesamtheit der Informationen betrachtet, spricht man von der DIB, der Directory Information Base - der oben angesprochenen Datenbasis. Weitere wichtige Begriffe sind der RDN, der DN und der DIT. Dabei handelt es sich um den (Relative) Distinguished Name und den Directory Information Tree. Alle Objekte werden in einer Baumstruktur abgelegt, da beim Entwurf die Meinung vorherrschte, dass die Realität am ehesten als solche abgebildet werden kann. Dieser Baum ist der DIT. Jedes Objekt in diesem Baum hat innerhalb seiner Hierarchieebene einen eindeutigen Bezeichner, den RDN. Dieser bezeichnet das Objekt exakt relativ zu seinem Vorgänger. Der DN wiederum gibt den vollständigen Bezeichner eines Objektes in der Gesamtstruktur an.

Am Beispiel bedeutet dies, dass „MarkoDamaschke“ ein eindeutiger RDN in den Reihen der Informatik-Studenten des Jahrgangs 2000 ist, aber nicht weltweit sein muss. Nimmt man hingegen etwa „Deutschland“ → „TUChemnitz“ → „Informatik – Fakultät“ → „S2000 – Studenten“ → „MarkoDamaschke“ ist dies auch im weltweiten Zusammenhang eindeutig und damit der DN.

Das X.500-System besteht aus einer Client/Server-Architektur, die mittels ISO-OSI verbunden ist. Der Client heißt DUA (Directory User Agent), der Server DSA (Directory System Agent). Zur Kommunikation kommen mehrere Protokolle zum Einsatz. Zwischen dem DUA und seinem nächsten DSA wird DAP, das Directory Access Protocol, gesprochen, zwischen den einzelnen DSA eines verteilten Verzeichnisses kommen DSP, das Directory System Protocol, aber auch DOP, Directory Operational Binding Management Protocol, und DISP, Directory Information Shadowing Protocol, zum Einsatz.

Das DAP ist sozusagen die Abfragesprache des Verzeichnisnutzers. Wird dabei eine Information gewünscht, die dem angesprochenen DSA nicht vorliegen, fragt er mittels DSP bei anderen DSA des Verzeichnisses an. Dabei kann er natürlich alle ihm bekannten DSA anfragen, kann aber auch, wenn ihm Informationen über die Verteilung des DIT vorliegen, gezielt vorgehen. Zur Verbreitung dieser Informationen wird zwischen den DSA das DOP eingesetzt. Das dritte Protokoll zwischen den Servern dient der Replikation der Verzeichnisdaten, um die Ausfallsicherheit zu erhöhen und die Reaktionszeit des Verzeichnisses auf Anfragen zu verringern. Das DAP kennt eine Anzahl von Operationen, die dem Binden ans Verzeichnis, dem Lösen von selbigen, dem Lesen eines Verzeichniseintrags, dem Vergleichen eines Attributwerts mit einer Eingabe, dem Suchen bestimmter Attributmuster, dem Anlegen und Löschen von Einträgen, dem Umbenennen von RDNs und dem Verschieben von Einträgen im DIT durch Änderung des DN's entsprechen.

Die Struktur eines X.500-Verzeichnisses kann grob noch einmal in Abbildung 2.2 eingesehen werden.

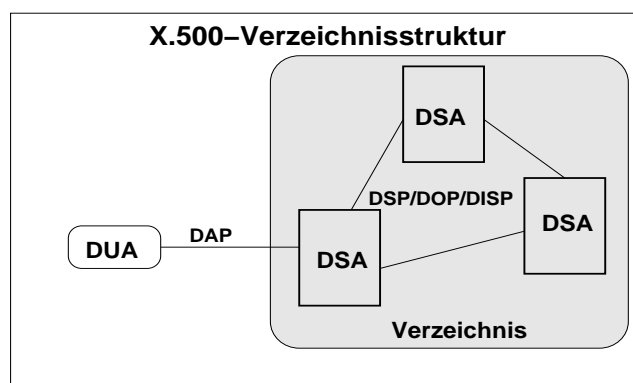


Abbildung 2.2: Struktur eines X.500-Verzeichnisses

1988 wurde die erste Version des Standards herausgegeben. 1993 und 1997 folgten Erweiterungen. Dabei wurden jeweils Verbesserungen eingebaut, die auf Erfahrungen des bisherigen Standards beziehungsweise neuen Gegebenheiten basierten. So war anfangs nur ein Modifizieren von Blatteinträgen möglich, was bei Restrukturierungen innerhalb von Organi-

sationen erheblichen Aufwand bei der Anpassung des Verzeichnisses bedeutete. Außerdem wurden Erweiterungen bei der Authentifizierung vor der Bindung ans Verzeichnis eingeführt und auch DOP und DISP wurden erst 1993 zur Replikationsstandardisierung erwähnt.

2.1.2 LDAP

Schon der Name LDAP, ausgesprochen Lightweight Directory Access Protocol, erinnert an die Entwicklungsnähe zu X.500. Diesem ISO-Standard wurde oft seine Komplexität auf Grund der Anlehnung an den OSI-Standard angelastet. In den Anfangszeiten war damit ein gewaltiger Mehraufwand verbunden, denn die üblichen Arbeitsplatzrechner waren mit Hilfe der PC-Netze verbunden, die mit NetBIOS oder IPX/SPX arbeiteten, gelegentlich dem aufkeimenden TCP/IP. LDAP war dabei der Versuch, die Vorteile eines offenen Verzeichnisdienstes mit der einfacheren Netzinfrastruktur des TCP/IP-Protokollstacks zu verbinden. LDAP war dabei nicht der erste Versuch, ist aber aus heutiger Sicht der erfolgreichste. Weitere Vertreter waren der Directory Assistance-Dienst mit seinem Frontend fred und der Serverkomponente dad, aber auch DIXIE der direkte Vorgänger von LDAP.

LDAP war in seinen ersten Ausprägungen als Middleware zwischen einem X.500-DSA und einem auf TCP/IP-basierenden Klienten gedacht. Allgemein stand LDAP als Protokollkonverter zwischen einem TCP/IP-Klienten und verschiedenen Verzeichnissen - neben X.500 in LDAPv3 auch zum Beispiel Lotus Notes, MS Exchange oder Novell Directory Service. Die Platzierung in der Kommunikation liegt also zwischen dem Nutzer-Agenten und dem Verzeichnisserver, wobei sein Betrieb sowohl auf dedizierten Servern aber auch auf dem Verzeichnisserver angedacht war. Abbildung 2.3 verdeutlicht diesen Einsatz grafisch.

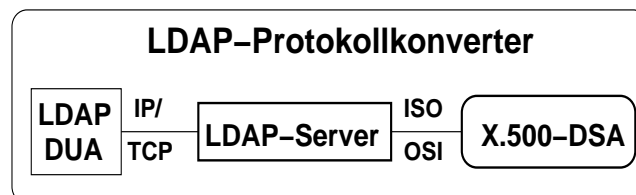


Abbildung 2.3: LDAP als Protokollkonverter

Aufgrund dieser Vorstellung eines LDAP-Dienstes unterscheidet [MR99] auch noch zwischen einem LDAP-Server und den SLAP-Servern. Erstgenannte sind diese Middleware-Protokollkonverter, während letztere für Standalone LDAP Server stehen und LDAP-Server mit eigenem Backend bezeichnen. Erste SLAP-Implementierungen entstanden in einem späten Stadium von LDAPv2 und waren in LDAPv3 dann Teil der Vereinbarung. Da LDAP auf Bestreben und in der ASID-Arbeitsgruppe (Access, Searching and Indexing of Directories) der IETF entstand, ist es in RFCs vereinbart, nicht wirklich standardisiert. Die Reference-Implementierung eines SLAP-Daemons entstand an der University of Michigan, die stark an der Erarbeitung der RFCs beteiligt war und deren Entwicklungen später in das OpenLDAP-Projekt (siehe [Ope05b]) übergingen.

Mit der wachsenden Verbreitung von SLAP-Servern entfiel zunehmend die Unterscheidung zwischen LDAP als Protokollkonverter und einem eigenständigen Verzeichnisdienst-Server.

So findet sich in [Car03] zwar immer noch die Erwähnung, dass LDAP die Vereinfachung von DAP im Sinne des Protokollstacks darstelle, ergänzt um die Reduzierung auf 9 wesentliche Operationen des DAP-Protokolls, die Unterscheidung von LDAP als Konverter beziehungsweise als eigenständiger Server findet sich dort allerdings nur noch indirekt, in dem gesagt wird, dass LDAP keinerlei Aussage über die Datenablage trifft, ausschließlich über den Datenzugriff. Vielmehr vertieft der Author sich stark in die Erläuterungen des aktuellen Stands des OpenLDAP-Projekts und die administrativen Feinheiten derer SLAP-Serverkomponente.

Ein weiterer Aspekt, der mit LDAPv3 eingeführt wurde, ist das in RFC 2849 definierte LDIF, das LDAP Interchange Format. LDIF ist ein textbasiertes Format, welches dem Austausch von Daten zwischen verschiedenen Verzeichnissen dient. Jede Datei dieses Formats enthält 3 wesentliche Merkmale, eine Sammlung von Verzeichniseinträgen, jeweils durch eine Leerzeile getrennt, einer Zuordnung von Werten zu Attributnamen und Anweisungen für den LDAP-Parser zum Umgang mit den Daten. Dieses Format kann eventuell in der späteren Betrachtung interessant sein, um Informationen in das AD zu importieren.

Wie bei jedem Dienst stellt sich auch bei einem Verzeichnisdienst die Frage der Authentifizierung und Autorisierung. Ein Verzeichnisdienst bietet, wie bereits anfangs angesprochen, die Möglichkeit einer dezentralen Datenpflege. Dabei kann beispielsweise vorgesehen werden, dass jeder Nutzer seine Personen gebundenen Daten selbst pflegt oder Abteilungen für ihre eigenen Datenbestände zuständig sind. Außerdem ist zu regeln, wer Zugriff auf welche im Verzeichnis abgelegten Informationen bekommt. Dazu wird bei LDAP die Operation der Bindung genutzt und je nach Implementierung bieten sich verschiedene Sicherheitsmechanismen. OpenLDAP bietet unter anderem einen anonymen Zugriff, den Zugriff mittels Nutzer und Passwort, wobei dies beides Informationen sind, die im Verzeichnis enthalten sein müssen und sowohl unverschlüsselt oder per SSL/TLS-Sicherung übertragen werden können, und die Anbindung an die SASL-Funktionalität. SASL ist eine Erweiterung, die die Anbindung verschiedener Authentifizierungsmechanismen über eine einheitliche Schnittstelle an ein System ermöglicht. So ist darüber auch eine „Kerberisierung“ des OpenLDAP erreichbar.

Ebenfalls ist LDAP wie X.500 als verteiltes Verzeichnis angelegt. Gründe für die Verteilung finden sich in der Performance, der geografischen und der administrativen Verteilung. Die Verteilung wird im Standard durch zwei Referenz-Zeiger gelöst, einer subordinate- und einer superior-knowlege-reference, die auf den Rechner mit dem Teilbaumwissen über darunter liegende beziehungsweise darüber liegende Informationen verweisen.

Da das Leistungsangebot von LDAP mit dem von X.500 vergleichbar ist, dabei die starke Nähe zu TCP/IP besteht und mit dem Siegeszug des Internets die Verbreitung auch von LDAP stieg, stellt LDAP zunehmend auch in Umgebungen großer Datendienstleistungen eine Konkurrenz dar und wird weithin als der Standard für Verzeichnisse angesehen. Gerade das Aufkommen der eigenständigen SLAP-Server unterstützte diese Tendenz. So hat sich Microsoft mit der Einführung eines Verzeichnisdienstes nach X.500 mit einer LDAP-Standardschnittstelle diesem Trend angeschlossen. Näheres findet sich dazu in Kapitel 4.

2.2 Authentifizierung

Moderne Mehrbenutzerbetriebssysteme oder vernetzte Dienste sollen in ihrer Nutzbarkeit administrativ auf einen sinnvollen Personenkreis von Nutzern eingegrenzt werden. Dazu muss im ersten Schritt sichergestellt werden, dass die Person, die den Dienst oder das System nutzen will, auch derjenigen entspricht, die sie vorgibt zu sein. Diesen Schritt nennt man die Authentifizierung. Im Anschluss kann dann anhand der Information, welche Person tatsächlich einen Dienst nutzen will, entschieden werden, ob sie dies darf. Dieser Schritt heißt wiederum Authorisierung.

Es gibt verschiedenste Konzepte, die zur Authentifizierung genutzt werden können. Die meisten bauen auf der Überprüfung des Wissens um ein gemeinsames Geheimnis auf. So ist das verbreitetste Verfahren die Eingabe eines Nutzernamens und eines Passworts. In anderen Situationen kommen SmartCards, Secure-ID-Generatoren, biometrische Daten oder eine Kombination aus verschiedenen Systemen zum Einsatz – die Vertraulichkeit der dadurch geschützten Daten, Systeme oder Dienste und der vertretbare Aufwand für deren Schutz ist hier für die Wahl ausschlaggebend.

Die Information über das gemeinsame Geheimnis oder die zur aktuellen Generierung notwendig ist, muss im System gespeichert werden. Im Standard-Linux-System wird hierfür die Datei */etc/passwd* genutzt, in älteren Windows-Systemen war es der Security Accounts Manager (SAM). Natürlich soll gerade im Umfeld dieser Arbeit auch nicht unerwähnt bleiben, dass die Verlagerung dieser Information in ein Verzeichnis möglich ist. Neben Windows-2000-Domänen (und folgend), wo die Nutzer-Informationen im Active Directory und damit in einem Verzeichnis liegen, kann man auch bei den verschiedenen UN*X-Systemen eine Anbindung an beispielsweise ein LDAP-Verzeichnis vornehmen. Eine andere Möglichkeit bestand zum Beispiel im Network Information Service (NIS), welcher allerdings nach heutigen Maßstäben nicht mehr als sicher genug und damit empfehlenswert angesehen werden kann.

Ein zunehmendes Problem in verteilten Systemen stellt ebenso die Vielzahl der Stellen dar, in denen Nutzerinformationen gespeichert und zu denen die Informationen (oft unverschlüsselt) übertragen werden und die Vielzahl an unterschiedlichen Passwörtern, die ein Nutzer sich einprägen muss, arbeitet er in verteilten System unterschiedlicher Administration. Auch die wiederholte Eingabe von Passwörtern bei der Arbeit kann sich als störend erweisen.

2.2.1 Das Kerberos-Protokoll

Die eben genannten Probleme waren die Hauptgründe der Entwicklung von Kerberos; die zentrale Authentifizierung in einer verteilten Umgebung, ein Single-Sign-On und die Vermeidung der Übertragung von Klartext-Passwörtern.

Kerberos ist ein Produkt des Athena-Projekts, welches ursprünglich den Einsatz von Computern und Software im Alltag im Massachusetts Institute for Technology (MIT) koordinieren sollte. Ein anderes Produkt dieses Projekts war beispielsweise das X Window System. Die Weiterentwicklung und Vereinheitlichung des Kerberos-Standards wird derzeit in einem Projekt mittels Declarations vorangetrieben, deren Internet-Portal auf [Ker05] zu finden ist. Der Name Kerberos wurde der griechischen Mythology entliehen und bezieht sich auf den

Wächter des Hades, der den Zugang zur Welt der Toten bewachte und Lebenden diesen versagen musste – sozusagen ein historisches Zugangssystem darstellte.

Kerberos ist ein Single-Sign-On-System, da nach einmaligem Anmelden am Kerberos-System eine Nutzung aller „kerberisierten“ Dienste möglich ist. Kerberos stellt durch seine zentrale Position eine Vermittlung zwischen unterschiedlichen administrativen Diensten dar – ein Nutzer kann Dienste eines anderen administrativen Bereichs nutzen, wenn der dortige Administrator dies erlaubt, da der Kerberos-Dienst gesichert die Nutzer authentifiziert. Außerdem gilt Kerberos aus zwei Gründen als sicher: Erstens wird nicht nur der Nutzer darauf überprüft, ob er derjenige ist, der er vorgibt zu sein, sondern auch die Kommunikationspartner also die Computer. Zweitens werden nur verschlüsselte Informationen übertragen und selbst wenn ein Angreifer ein Ticket für sich nutzbar machen kann, ist dieses zeitlich in seiner Gültigkeit begrenzt und damit die Sicherheit auf lange Sicht gegeben. Diese Aussage ist natürlich immer mit der Notwendigkeit verknüpft, dass die Nutzer sichere Passworte wählen und diese nicht ausspionieren lassen – wie bei den meisten Systemen ist der Nutzer das schwächste Glied in der Kette.

Die wesentlichen Begriffe im Zusammenhang mit Kerberos lauten:

1. Realm,
2. Principal,
3. Instance,
4. Ticket,
5. Keys, Salt und
6. Key Distribution Center (KDC).

Wie die Übersetzung des ersten Begriffs schon besagt, definiert der Realm ein Hoheitsgebiet, hier natürlich ein administratives. Ein Realm ist ein Namensraum, der sich eindeutig von allen anderen weltweiten Kerberos-Installationen unterscheidet. Dazu gilt die Vereinbarung, dass der durchgängig großgeschriebene DNS-Domänenname der Realm-Bezeichnung entspricht. Die Realm-Bezeichnung ist case-sensitive. Der Kerberos-Server beziehungsweise das Key Distribution Center (KDC) des Realms ist der zentrale Administrationspunkt für das gesamte Zugangssystem aller dem Realm angeschlossenen Systeme; somit ein sehr sensibler Punkt.

Das KDC stellt somit in der Betrachtung eines verteilten Dienstes nach Client-Server-Modell den Server dar. Die Principals sind die Klienten.

Jeder Nutzer, jeder Rechner und jeder Dienst eines Realms bekommt einen eigenen Bezeichner und einen zugeordneten Schlüssel. Diesen Bezeichner nennt man Principal. Der Schlüssel ist das gemeinsame Geheimnis; mehr dazu in der Folge. Der Principal wiederum ist weltweit eindeutig, was dadurch begründet ist, dass der Bezeichner innerhalb des Realms eindeutig sein muss und der Realm ebenfalls eindeutig ist, somit auch der Gesamtkonstrukt aus Realm und zugehörigem Principal eindeutig ist. Eng verknüpft damit ist der Begriff der Instance, die eine spezielle Ausprägung eines Principals bezeichnet. Am Beispiel eines Nutzers „Max Mustermann“ im Realm „BEISPIEL.REALM“ kann dieser das Principal

mmax@BEISPIEL.REALM haben. Ist er aber auch mit administrativen Aufgaben betraut, kann er zusätzlich auch die Instance *mmax/admin@BEISPIEL.REALM* besitzen.

Zwischen allen beteiligten Partnern einer Kerberos-Kommunikation werden ausschließlich verschlüsselte Datenpakete ausgetauscht, die kryptografische Informationen zur Authentizitätskontrolle der Kommunikationspartner enthalten. Diese Informationspakete heißen bei Kerberos Tickets. Es dient als eine Art Ausweis dem Dienst gegenüber, der genutzt werden soll.

Schließlich bleibt noch der Begriff des Salts. Jedem Principal ist wie bereits erwähnt ein Key zugeordnet, also ein gemeinsames Geheimnis zwischen Principal und KDC. Dieser Key ist ein kryptografischer Schlüssel, also das Ergebnis einer Transformations- und Hashingfunktion. Diesen kann sich natürlich kein menschlicher Nutzer merken und bei jedem Login eingeben, weshalb dieser Key aus seiner Passwort-Eingabe transformiert wird. Um die Stärke der Verschlüsselung zu erhöhen, wird vor der Transformation eine Zeichenkette an die Eingabe angehängt. Diese Zeichenkette nennt man Salt, also Salz. Kerberos V5 beispielsweise hängt standardmäßig den Realm-Namen an, was verhindern soll, dass einem Nutzer bei Verwendung gleicher Passworte in unterschiedlichen Realms der gleiche Key zugeordnet wird.

Um sicherzustellen, dass keine Passworte übertragen werden, wurde ein Protokoll für Kerberos entwickelt, welches auf dem Needham-Schroeder-Protokoll basiert. Dieses Protokoll baut darauf auf, dass eine zentrale Stelle mit Kenntnis über alle Schlüssel aller Principals die Authentizität aller Kommunikationspartner sicherstellt. Diese zentrale Stelle ist das Key Distribution Center (KDC), welches aus zwei Komponenten besteht:

1. dem Authentication Server (AS) und
2. einem Ticket Granting Server (TGS).

Die Erteilung eines Zugangs zu einem Dienst oder System geschieht bei Kerberos in 2 Schritten. Erst wird für ein Realm, in dem das System oder Dienst angesiedelt sind, ein Kerberos Ticket Granting Ticket (KRBTGT) abgefragt. Dies ist für die Dauer seiner Gültigkeit der Nutzausweis, also ein zeitlich begrenztes Sitzungsvisum. Mit diesem durch den AS erteilten Ticket kann der Client beliebige Service Tickets beim TGS anfordern. Das KRBTGT ist sozusagen die Überprüfung der Authentizität des Principals, die Service Tickets sichern die Authentizität der Kommunikationspartner und Principals untereinander.

Um ein solches KRBTGT zu erlangen, sendet der Client sein Principal und den Namen des KRBTGT-Principals, welches er anfordert, kombiniert mit Zeitinformationen an den AS. Dieser erteilt, wenn alle beteiligten Principals in seiner Datenbank bekannt sind, ein Ticket, in dem ein Session-Key einmal für den TGS, verbunden mit Informationen zum anfragenden Client-Principal, und einmal für das Client-Principal jeweils mit deren Key verschlüsselt enthalten ist. Dadurch kann der echte Client-Principal mit Hilfe seines Passworts seinen Session-Key entpacken. Der im zweiten Schritt angefragte TGS bekommt vom Client den mit seinem Schlüssel kodierte Teil des Tickets übermittelt, in dem ja auch Informationen zum Principal enthalten sind, der das KRBTGT bekommen hat. Auf dieser Grundlage kann der TGS die Authentizität des übermittelten Session-Keys und des übermittelnden Principals überprüfen. Im Ergebnis wird der TGS ein Service-Ticket erteilen, welches einen Session-Key für den Service inklusive Informationen zum ermächtigten Principal enthält

und mit dem Schlüssel des angefragten Dienstes verschlüsselt ist. Dieses Ticket wird natürlich auch verschlüsselt übermittelt, diesmal wiederum mit dem Session-Key, der in der AS-Kommunikation ausgeteilt wurde.

In Kerberos V4 standen Probleme im Protokoll, welche einen sicheren Einsatz in heutigen Umgebungen nicht mehr vollständig gewährleisten. So war eine feste Bindung an Single-DES als kryptografische Funktion ebenso hinderlich, wie keine Bindung an eine Byte-Order, sondern ein Flag in der Kommunikation, das in jeder Nachricht deren Byte-Order anzeigte. Kerberos V5 wurde daher grundlegend neu entworfen unter Einfluss der Erkenntnisse aus dem Kerberos V4-Einsatz.

Die nennenswerten Erweiterung sind in erster Linie die Einführung eines beliebigen Verschlüsselungsverfahrens, sogar innerhalb einer Sitzung. Unterstützt so beispielsweise das Client-Principal nur Single-DES, der angefragte Dienst aber Triple-DES kann das KDC die Tickets in unterschiedlichen Verfahren verschlüsseln, solange sichergestellt ist, dass die beteiligten Kommunikationspartner jeweils ihren Teil des Tickets entschlüsseln kann. In Kerberos V4 wurde das verschlüsselte Ticket für den angefragten Dienst (beispielsweise auch der Teil eines KRBTGT, welches dem TGS als Ausweis des Principals dient) ebenfalls in die Verschlüsselung des Principals einbezogen. Dies erwies sich als unnötig, da das Geheimnis auch durch die Verschlüsselung für den Zieldienst bei einer Übertragung gewahrt werden kann. Somit wird in Kerberos V5 das Ticket aus zwei getrennt verschlüsselten Teilen übermittelt. Auch das vereinfacht den Einsatz unterschiedlicher Kodierungsverfahren in den Ticket-Teilen.

Außerdem wurden Optionen für Tickets eingeführt. So können Tickets eines Nutzerprincipals auf andere Rechner verschoben werden (forwardable tickets und proxiable tickets – ermöglicht die Anforderung eines neuen TGTs beziehungsweise nicht) oder der Gültigkeitszeitstempel kann erneuert (renewable ticket) beziehungsweise für einen zukünftigen Zeitpunkt angefordert (postdated ticket) werden.

Ebenfalls mit Kerberos V5 wurde das Prinzip der Pre-Authentication eingeführt. Pre-Authentication soll die Daten der Kerberos-Datenbank gegen offline Wörterbuch- und brute-force-Attacken besser schützen. In Kerberos V4 wurde durch den AS jedem anfragenden Client ein TGT für ein beliebig angefragtes Principal erteilt. Dieses war zwar mit dem Schlüssel des angefragten Principals kodiert, aber damit gegen oben genannte Angriffe verwundbar. Bei der Pre-Authentication wird seitens des Client-Principals ein Zeitstempel mit dem Key des Principals verschlüsselt und an den Request zum AS angehängt. Dieser kann mit dem Key des anfragenden Principals dessen Zeitstempel entschlüsseln und mit seiner Zeiteinstellung vergleichen. Innerhalb einer administrierten Zeittoleranz wird der Principal als „echt“ anerkannt und das angefragte TGT erteilt.

Eine weitere sehr interessante Eigenschaft von Kerberos sind so genannte „Cross Realm Trusts“. Dabei handelt es sich um Vertrauensstellungen zwischen mehreren administrativen Hoheitsgebieten. Dadurch wird es ermöglicht, Ressourcen fremder Realms für Nutzer des eigenen zugänglich zu machen. Dazu werden in den Kerberos-Datenbanken jeweils zwei Principals eingepflegt, die den Mustern *krbtgt/EIGENER.REALM@FREMD.REALM* und *krbtgt/FREMD.REALM@EIGENER.REALM* entsprechen und jeweils mit dem selben Schlüssel in beiden Datenbeständen erstellt sind. Dadurch kann jeder Nutzer ein „Intermediate Ticket Granting Ticket“ des Fremdrealms anfordern und mit diesem direkt beim Fremd-

TGS Service Tickets für die Ressourcen des Fremdrealms anfordern. Diese Form nennt sich direkter Cross-Realm Trust und ist in Kerberos V4 die einzige unterstützte Form. In Kerberos V5 wurde zusätzlich das Prinzip des „certification path“ eingeführt und damit der impliziten oder transitiven Vertrauensstellung. Der direkte Trust wird nur noch zu einem zwischengeschalteten Realm erstellt, der wiederum anderen Realms vertraut. Dadurch wird implizit auch eine Vertrauensstellung zu dem dritten Realm über den zwischengeschalteten Realm aufgebaut. In einem Windows-Domänen-Baum wird durch die Root-Domäne automatische solch eine Vertrauensstellung aufgebaut. Mehr dazu in Kapitel 4.

In allen Definitionen von Kerberos fehlt eine Festlegung zur Änderung des Passworts eines Nutzers selbst. Der Moment der Änderung bedarf als einziger der Übertragung eines kryptografischen Geheimnisses – entweder des Klartextpasswort oder des neuen Schlüssels. Dazu wird implementationsabhängig ein Dienst genutzt, der ein externes Protokoll nutzt und meist das Ergebnis der Kodierung des neuen Passworts überträgt. Dadurch wird auch dabei kein Klartextpasswort übertragen. Die Protokolle in Kerberos V4, die dabei zum Einsatz kamen, waren die administrativen Protokolle des jeweiligen Herstellers. In Kerberos V5 kommt nun hingegen meist das Horowitz-Protokoll zum Einsatz, allerdings oft auch noch mit implementationsabhängigen Anpassungen. [Gar03] erwähnt noch die Einführung eines einheitlichen Protokolls, welches auf Horowitz aufsetzt, aber Nutzern das Ändern des eigenen und Administratoren zusätzlich das Ändern von Fremdpasswörtern ermöglicht. Dieses befand sich 2003 noch in der Phase der Definition, wodurch bisher noch kein Einsatz stattfindet. Ein Problem bei den Verfahren, bei denen seitens des Klienten die Transformation vorgenommen wird, liegt in der leistungsstarken Überprüfung der kryptografischen Stärke des neuen Passworts.

2.2.2 Alternativen zu Kerberos

Andere Produkte mit ähnlichen Zielen sind unter anderem das auf [DCE05] verbreitete DCE, das Distributed Computing Environment, welches verschiedene Werkzeuge für ein verteiltes System bereitstellt. Die Zugangskontrolle ist Kerberos V5-basiert und daher weniger als Alternative interessant.

Das Authentifizierungssystem des Globus Toolkits, das GSI, die Globus Security Infrastructure arbeitet mit ähnlichen Zielen, aber auf Basis der Public-Key-Verschlüsselung und einer Zertifikatinfrastruktur. Allerdings wurde Globus für HPC-Umgebungen entworfen und hat damit eine etwas andere Ausrichtung. Aber die Entwickler von Globus haben die Notwendigkeit der Interoperabilität mit Kerberos erkannt und haben Translatoren zwischen Kerberos Tickets und GSI Zertifikaten entwickelt. Genauere Informationen finden sich unter [Glo05]. Ein letztes Projekt ist auf Anregung der Europäischen Kommission entstanden und nennt sich „Secure European System for Applications in a Multivendor Environment“ oder kurz SESAME. Die Technologie ist gleich derer von Kerberos, sogar kompatibel, aber es bestehen Verbesserungen beispielsweise in der Kontrolle des Nutzers über die von ihm verteilten Informationen. Die Verteilung der Informationen zum Projekt erfolgt über [Ses05].

2.3 Authorisierung

Der Mechanismus, der zur Authorisierung eingesetzt wird, ist üblicherweise der Abgleich einer Information, die dem zugreifenden Objekt zugeordnet ist, mit einer Information, die dem Objekt zugeordnet ist, auf welches zugegriffen werden soll, und bestimmt, welche Objekt zugreifen dürfen.

Eine weitverbreitete Variante ist die der ACLs, der Access Control Lists. Dabei wird einem zu sicherndem Objekt, beispielsweise einem Verzeichnis im Filesystem, eine Liste zugeordnet, die wiederum eine Liste von Objekten, beispielsweise Nutzerkennzeichen, enthält und den zugehörigen Berechtigungen, hier etwa das Anzeigen, Lesen, Schreiben, Löschen oder Anlegen von Dateien.

Eine andere einfache Variante wurde in UN*X-Systemen ursprünglich zur Nutzerauthorisierung eingesetzt. Dabei wurde in der Datei */etc/passwd* in der Zeile, die jedem Nutzerkennzeichen zugeordnet ist, mittels eines Ausrufezeichens statt eines Passworthashes angezeigt, dass der Zugriff auf das System für den Nutzer gesperrt ist. Und innerhalb des ext2-Filesystems sind die Dateirechte, also die Authorisierung hinsichtlich des Zugriffs auf die einzelne Datei, durch einen Abgleich mit dem Eigentümer und einer zugeordneten Gruppe jeweils mit dem Zugreifenden realisiert, denen dann jeweils das Recht zum Lesen, Schreiben und Ausführen mittels Bitvergleichs zugestanden wird.

Andere Lösungen auf Basis von Datenbanken, Text-Dateien oder binären Datenhalden sind denkbar.

Bei NTFS wird diese ACL, genauer gesagt zwei ACLs, im SID-Container des Fileobjekts abgelegt. Eine ACL, die DACL, bestimmt die Zugriffsrechte auf das File, die SACL regelt die Benachrichtigungsereignisse bei Erfolg und Mißerfolg des Zugriffs. Bei sind über den ACL-Editor beziehungsweise den „Sicherheit“-Tabulator der Eigenschaften des Objekts anzupassen.

3 Ausgangslage und Nutzungsszenarien im URZ

Da im Rahmen der Arbeit eine Erweiterung des Dienste-Spektrums untersucht werden soll, ist es am Anfang notwendig, das bisher bestehende Angebot aufzuzeigen und näher zu beleuchten.

Dienste, die im Umfeld einer Betrachtung von Active Directory interessant sind, sollen in der Folge in ihrer aktuellen Ausprägung im URZ vorgestellt werden.

Zu allererst sollen allerdings Szenarien betrachtet werden, die für einen Einsatz eines Active Directorys im Universitätsrechenzentrum der TU Chemnitz sprechen.

3.1 Einsatzszenarien

Wie bereits angesprochen, ist das Active Directory die zentrale Komponente im Konzept von Microsoft zur Verwaltung von Windows-Server basierten Netzen.

Die an verschiedenen Stellen dieser Arbeit angesprochenen Struktureinheiten können sowohl Lehrstühle der Universität, zentrale Einrichtungen, aber auch Forschungsprojektgruppen sein.

3.1.1 Einsatz von Windows-Servern in Struktureinheiten

Diese Struktureinheiten wollen verschiedene Software einsetzen, die von Microsoft hergestellt werden oder das Vorhandensein eines Active Directorys voraussetzen. Andererseits wollen Sie Informationsquellen nutzen, die bereits im URZ vorhanden sind oder weitere Dienste dieses verwenden oder auf einfache Weise zusammenarbeiten und ihre Ergebnisse austauschen.

Auf Grund der Verbreitung oder Notwendigkeit wird in solchen Fällen oft ein eigener Windows-Server in den Struktureinheiten vorgesehen oder angedacht. Und damit auch der Betrieb einer eigenen Active Directory Domäne geplant, wodurch alle Rechner der Domäne die Ressourcen des Windows-Servers nutzen können, wie etwa Laufwerksfreigaben, die dann als Projektlaufwerke dienen, oder Drucker der Struktureinheit oder Software, die über den Server bereitgestellt wird, oder die Verwaltung der lehrstuhleigenen Arbeitsplätze. Dabei ist es wünschenswert, möglichst viele Informationen aus dem zentralen Pool zu beziehen und nur die zu erweitern, die projektbezogen sind. Da ein Windows-Server, der als Domaincontroller eingesetzt wird, verschiedene Dienste und Einstellungen voraussetzt, sind die derzeit betriebenen AD-Domänen so genannte Insellösungen, die abseits jeglicher Integration in das URZ betrieben werden. Ein wesentlicher Punkt dabei ist der Namensraum im DNS. DNS ist das Werkzeug, welches Active Directory zum Auffinden von Ressourcen nutzt. Auf dem Domänencontroller wird der Name der Domäne direkt mit dem DNS-Suffix des Controllers verbunden, so dass in der selben DNS-Subdomäne jeweils nur eine Active Directory Domäne realisierbar ist. Die bestehende Netzstruktur macht daher beispielsweise den Betrieb zweier AD-Domänen innerhalb einer Fakultät unmöglich. Daher wurde auf die oben erwähnten

Insellösungen ausgewichen. Diesem Trend soll durch Bereitstellung eines zentralen Active Directorys mit Anbindung an eine zentrale Nutzerverwaltung und Zugangskontrolle entgegengewirkt werden. Möglichst viele Informationen der zentralen Nutzerdatenbank (MoUSE) sollen in das zentrale AD dupliziert werden und damit allen Windows-Server-Umgebungen der TU Chemnitz zur Verfügung stehen. Außerdem wird eine Anbindung an das zentrale Kerberos V5-System angestrebt, wodurch allen Nutzern mit ihrem zentralen Passwort Zugang zu den Windows-Umgebungen eingerichtet werden kann. Dadurch gibt es keine Notwendigkeit alle Nutzerinformationen mehrfach zu pflegen und Nutzer bleibt erspart, sich verschiedene Passworte zu merken.

3.1.2 Vertrauenstellung zwischen Struktureinheiten

Auf Grund der Abstützung der Windows-Authentifizierung auf das Kerberos-System lassen sich weitere Vorteile umsetzen. Eine entscheidende Erweiterung ist die implizite Vertrauensstellung aller Windows-Domänen in einem Baum. Auf Grundlage des Vertrauens zur Root-Domäne, das transitiv und auch gegenseitig aufgebaut wird, lassen sich Vertrauensstellungen zwischen allen Windows-Domänen einrichten.

Die Folge dieses Vertrauens ist die Möglichkeit, Ressourcen wie beispielsweise Netzlaufwerke einer Domäne in verschiedenen anderen Domänen des Active Directory Baumes zu nutzen. Der Administrator einer Domäne kann dabei natürlich den Zugriff steuern, es bietet sich aber die Möglichkeit auf Grund der einheitlichen Nutzerverwaltung schnell und unkompliziert Nutzern Rechte in der eigenen Domäne einzurichten oder zu entziehen. Dadurch ließe sich lehrstuhlübergreifende Zusammenarbeit vereinfachen.

3.1.3 Bereitstellung von AD-basierter Software

Viele Produkte des Hauses Microsoft und auch stark windowslastiger Hersteller setzen seit der Einführung von Active Directory stark auf dessen Einsatz und Dienstbereitstellung. Verschiedene Produkte nutzen das Verzeichnis als Ablage von Konfigurationsinformationen oder Nutzerdaten beziehungsweise Einstellungen. Ein typischer Vertreter ist die Microsoft Groupware-Lösung „Exchange“. Der Betrieb einer solcher Software ist nur mittels Bereitstellung eines Active Directorys möglich. Die Bereitstellung des zentralen AD-Dienstes soll hier einerseits die Möglichkeit eröffnen, diesen Dienst als Basis solcher Software zu nutzen, andererseits indirekt durch den Anschluss einer Subdomäne helfen, die dadurch zentrale Informationen nutzen kann und dann zur Bereitstellung dieser Software dient.

3.1.4 Active Directory als zentraler Verzeichnisdienst

Ein weiterer Aspekt, der den Einsatz eines ADs begünstigt, ist die laut Microsoft genannte Standardkonformität, mit der auf das Active Directory beziehungsweise dessen Informationen mittels des LDAPv3-Protokolls zugegriffen werden kann.

Somit ist es denkbar, das Active Directory als den zentralen Verzeichnisdienst im Universitätsrechenzentrum zu nutzen. Schlußfolgernd ist es möglich, beispielsweise Mailprogramme an die Informationsquelle Active Directory anzuschließen. Auch viele andere Softwareprodukte setzen auf die Dienste eines LDAP-Verzeichnisses, in welchem Nutzerinformationen und Einstellungen der Produkte abgelegt werden. Es werden derzeit mehrere Groupware-Produkte hinsichtlich eines Einsatzes als Dienst im Universitätsrechenzentrum evaluiert. Je-

des setzt auf eine Datenbasis in einem Verzeichnis oder einer Datenbank. Dort werden neben Nutzerinformationen auch Adressen oder Kalenderdaten der Nutzer abgelegt.

Entspricht die Aussage von Microsoft hinsichtlich der Standardkonformität des LDAP-Interfaces in jeder Hinsicht der Implementierung, kann das AD auch hier eine wichtige Rolle einnehmen.

3.2 Aktueller Stand im URZ

3.2.1 Authentifizierung

Als zentrales Werkzeug der Authentifizierung wird Kerberos V5 in der OpenSource-Implementierung der Königlich Technischen Hochschule in Schweden „Heimdal“ eingesetzt. Hierbei wird aktuell die Version 0.6.3. genutzt, zeitgleich die neue Version 0.7. evaluiert, deren interessanteste Neuerung neben weiteren Verschlüsselungs- und Hashing-Algorithmen in der optionalen Verlagerung des Ticket-Caches auf dem Klienten in den Arbeitsspeicher liegt. Die jeweils aktuelle Version, ein täglicher CVS-Snapshot und der Zugang zu den Projektmailinglisten kann auf der Projekthomepage [Hei05] gefunden werden.

Für einen Überblick über die Funktionsweise und die wesentlichen Begrifflichkeiten im Zusammenhang mit Kerberos wird auf 2.2.1 verwiesen.

3.2.2 Authorization

Da dieses Thema von Dienst zu Dienst differiert, kann hierbei keine derart einfache Aussage getroffen werden, wie bei der Authentifizierung. Diese bildet natürlich die wesentliche Grundlage, um eine sichere Authorisierung durchführen zu können.

Zentral gespeichert und verwaltet werden die nutzerbezogenen Informationen, also auch die Berechtigungen (hier vor allem Gruppenzugehörigkeit zu typischen Nutzergruppen) in der MoUSE, einer MySQL-Datenbank, die speziell für die Nutzerverwaltung des URZ der TU Chemnitz entwickelt wurde. Weitere Informationen hierzu finden sich in [MoU01].

Eine weitere wesentliche Komponente sind hierbei Access Control Lists (ACLs) im AFS (Andrew File System), welches in der OpenAFS-Implementierung eingesetzt wird und einen kerberisierten Dienst darstellt, also vollkommen auf eine Authentifizierung durch Kerberos aufsetzt. Dabei werden Rechte nutzer- oder gruppenweise für einzelne Verzeichnisse oder ganze Verzeichnisbäume vergeben. Da das AFS als zentrales Filesystem zum Einsatz kommt, regelt sich über diese Verzeichnisrechte der Zugriff auf Daten aber auch Programmressourcen.

Auch der Zugang zu Rechnerkapazitäten wird im Wesentlichen auf Tokens des AFS zurückgeführt. Abgestützt wird sich im Linux-Umfeld hierbei auf ein PAM-Modul, bei Windows auf eine Gina-Bibliothek, welche jeweils versuchen, ein AFS-Token zu erhalten. Das AFS wiederum authentifiziert den Nutzer gegen das Kerberos-System. Bekommt der Nutzer ein gültiges AFS-Token, also ein KRBTGT- und ein AFS-Service-Ticket vom Kerberos, gilt er als authentifiziert und wird autorisiert. Weitere Informationen zu OpenAFS finden sich auf der Projekthomepage [Ope05a].

Im Linux-Umfeld wird weiterhin mittels der Verteilung von lokalen Konfigurationsdateien Einfluss auf die Authorisierung genommen. Diese Verteilung wird dabei durch dieselben Werkzeuge vorgenommen, die auch zur Softwareverteilung (siehe 3.2.5) dienen.

3.2.3 Domain Name Service

Jeder Active Directory Domäne wird eine Domain im DNS zugeordnet, deren Administration dynamisch durch den AD-Server und die beteiligten Clienten gemäß [DDN97] erfolgen soll. Außerdem ist es notwendig, dass die eingesetzte Software Resource Records zur Lokalisierung von Diensten, kurz DNS SRV, nach [DNS96] unterstützt.

Dazu wird in der Standardeinrichtung eines ADs vorgeschlagen, einen MS-DNS-Server auf dem AD-Server einzurichten, beziehungsweise ein angegebener DNS-Server auf seine Fähigkeit zu dynamischen Updates hin getestet.

Die im URZ eingesetzten Server arbeiten mit dem Berkeley Internet Name Domain (BIND) vom Internet System Consortium (ISC); siehe [BIN05].

Grundsätzlich ist dieses System in der Lage, allen Anforderungen gerecht zu werden, unterstützt also sowohl DNS SRV als auch DDNS, wobei Letzteres deaktiviert ist, um die administrative Hoheit des URZs zu wahren. Die Literatur zum Thema AD weist allerdings auch darauf hin, dass ein Betrieb grundlegend auch mit statischem DNS möglich ist, allerdings einen administrativen Mehraufwand und eventuelle Performance-Einbußen bedeutet.

3.2.4 Verzeichnisdienst

Es gibt im Servicespektrum einen LDAP-Server (mehr zu LDAP in 2.1.2), auf den Teile der MoUSE-Datenbank repliziert werden, so dass sich grundlegende Informationen der Nutzer im LDAP wiederfinden. Dies wird angeboten, um beispielsweise in Mailprogrammen (MUA) eine Namenssuche zu ermöglichen.

LDAP gehört nicht zu den zentralen Diensten und wird daher nur von einem einzelnen PC angeboten, wodurch keine Ausfallsicherheit geboten werden kann.

Andere Dinge, für die normalerweise der Einsatz eines Verzeichnisses angepriesen wird wie Authentifizierung und Autorisierung, werden wie eben erwähnt durch andere Werkzeuge angeboten.

Weitere typische Dienstangebote eines Verzeichnisses etwa Informationsgewinnung über Nutzer oder Nutzungsverhalten werden meist mittels Webfrontends zur Nutzerdatenbank (MoUSE) ersetzt.

3.2.5 Softwareverteilung

Ein Wunsch, der sich mit dem Einsatz von Active Directory verbindet, ist eine windowsintegrierte Variante der Softwareverteilung und -administration.

Bisher wird diese Aufgabe durch den Einsatz von CFEngine in Kombination mit Cygwin als Umgebung im zentralen Administrationsbereich des URZs substituiert, die Umsetzung in anderen Struktureinheiten ist individuell in deren Verantwortung. Mittels der CFEngine-Technologie werden msi-Pakete installiert oder Skripte per CMD-Batch beim Systemstart beziehungsweise nach einem festen Zeitplan ausgeführt. Dabei wird noch unterschieden nach bestimmten Klassen, in denen Rechner zusammengefasst werden, welche Skripte durch die CFEngine ausgeführt werden. Grundsätzlich werden die Pakete im AFS abgelegt und die Konfigurationsfiles, welche die zu installierenden Pakete und den Installationsvorgang beschreiben, werden durch die CFEngine auf die lokale Maschine kopiert und steuern dann von dort aus den Mechanismus. Lokal finden sich weiterhin Flagfiles, welche für jedes Paket die letzte installierte Version verzeichnen und somit zur Überprüfung der Notwendigkeit

eines Updates herangezogen werden können.

Auch wenn grundsätzlich innerhalb des CFEngine-Verständnisses ebenfalls eine Hierarchie besteht, wenn man die Gliederung in Klassen als solches verstehen will, ist es im Gegensatz dazu beim Einsatz von Active Directory absolut notwendig, dass die betroffenen Rechner in einer Windows Domain organisiert sind. Dabei ist außerdem zu beachten, dass ja im Falle von AD die Softwareverteilung auf Gruppenrichtlinien basiert und wie später in 4.7 erläutert wird, diese ebenfalls hierarchisch Anwendung finden, also eine Administration durch mehrere Stellen dieser Hierarchie Auswirkungen auf den betroffenen Rechner haben. Dies kann im schlimmsten Falle gar zu Inkonsistenzen führen, weshalb der Client widersprüchliche Konstrukte aus den Gruppenrichtlinien ausblendet. Auch hier ist jedes Endsystem für die Installation der bereitgestellten Pakete selbst verantwortlich und tut dies nur unter der Maßgabe der hierarchischen Vorgaben. Allerdings ist die Durchsetzung wegen der tiefen Verwurzelung der AD-Bestandteile im System einfacher. Bei CFEngine hingegen bleibt jedes Zielsystem ein autarkes, eine Durchsetzung durch eben externe Werkzeuge stellt sich deutlich schwerer dar und die individuelle Gestaltung des Zielsystems durch Zuordnung zu mehreren Klassen bleibt vielfältiger.

Auf Grund des massiven Einsatzes der CFEngine bereits im Umfeld der Administration von Unix/Linux-Systemen durch das URZ, dem damit verbundenen Know-How und der Integration durch CygWin besteht kein dringender Bedarf zur Anpassung des Verfahrens seitens des Rechenzentrums.

Anders sieht es im speziellen Fall der Installation beziehungsweise dem Upgrade des AFS-Clients sowie bei einzelnen Strukturbereichen aus. Beim Update des AFS-Clients entsteht der kritische Moment, dass der Dienst beendet werden muss, dass Paket aber aus dem AFS geliefert wird. Dabei kann der Einsatz von AD mit Hilfe von Gruppenrichtlinien eventuell Verbesserungen mit sich bringen. In den einzelnen Struktureinheiten findet sich häufiger das Know-How zur Administration auf Windowsbasis beziehungsweise schreckt die Möglichkeit via GUI zu administrieren weniger ab. Nebenbei bietet sich aber auch die Möglichkeit, grundlegende Administration durch das URZ anzubieten, bei gleichzeitiger Möglichkeit der individuellen Anpassung durch Struktureinheiten. Inwieweit Wechselwirkungen entstehen, die durch die gleichzeitige Anwendung von CFEngine- und Gruppenrichtlinien-Vorgaben auftreten, sollte in einem Test nach Aufbau einer zentralen AD-Struktur untersucht werden.

4 Active Directory im Detail

Eingehend sei zu bemerken, dass das Thema äußerst komplex ist und an dieser Stelle nicht endgültig und ausführlich besprochen werden kann. Einen ähnlichen Überblick dessen, was das Active Directory anbietet und ist, stellt der Hersteller auf [Act00] bereit. Vieles des dort Erwähnten findet sich in diesem Kapitel ebenfalls wieder, mit Betracht auf die Aufgabenstellung wird hier allerdings ein anderer Schwerpunkt gesetzt. Vertiefend sei auf die reichhaltige Literatur rund um das Thema verwiesen, welche nicht nur die Konzepte, sondern auch die Praxis sehr detailliert beschreiben. Dabei ist zu beachten, dass es Vertreter gibt, die wie [Sch03] oder [Mic03] nur randständig auf die Theorie eingehen, aber auch solche, die vom Konzept der Technologie zu klassischen Anwendungsfällen hinführen, so etwa [RA04]. [Sta03] faßt das große komplexe Thema recht oberflächlich, vielleicht auch einführend in folgenden Zeilen zusammen: „Die Betonung liegt bei Windows 2000 auf zusätzlichen Diensten und Funktionen für die Unterstützung der verteilten Verarbeitung. Das zentrale Element der neuen Merkmale von Windows 2000 ist eine Funktion mit dem Namen Active Directory, hinter der sich ein verteilter Verzeichnisdienst verbirgt, der in der Lage ist, Namen beliebiger Objekte beliebigen Informationen über diese Objekte zuzuordnen.“

Wie bereits mehrfach erwähnt, ist das Active Directory stark in die Windows Systeme seit Windows 2000 integriert. Der Dienst wird einerseits durch jeden Domänencontroller angeboten, ist sogar die Voraussetzung für den Betrieb als solcher. Das Verzeichnis ist die zentrale Datenablage zu allen Informationen rund um die Domäne. Erwähnenswert an dieser Stelle ist noch eine Windows-typische Eigenart der Verzeichnisstruktur. So wird jedem Objekt des Verzeichnisses eine eindeutige ID zugeordnet, die so genannte SID (Secure ID), welche einem Objekt beim Anlegen zugeordnet und nur für dieses gültig ist. Das bedeutet auch, dass ein gelöscht Objekt im Verzeichnisbaum, sollte es gelöscht und mit selbem Namen und Attributen erneut angelegt werden, eine neue SID erhält, welche nicht mit der bisherigen übereinstimmt.

Doch auch als Dienstanutzer ist Active Directory tief im Betriebssystem verankert. Einführend aber detaillierter zeigt dies [KT04], wobei sie klar stellt, dass es 2 zentrale Punkte gibt, in die Active Directory als Dienst auf den Windows-Clients eingreift. Davon befindet sich einer im Benutzermodus und einer im Kernelmodus. Das Sicherheitsteilsystem im Benutzermodus überprüft die Überwachung der Systemressourcen und der Anmeldeauthentifizierung und überwacht selbst die Benutzerkonten und die diesen zugewiesenen Zugriffsrechten. Außerdem wird die Wiederherstellungsrichtlinie durch AD durchgesetzt und abgesichert. Im Kernelmodus ist es der Sicherheitsreferenzmonitor, welcher die Durchsetzung der zugewiesenen Sicherheitsrichtlinien sicherstellen soll.

Intern besteht das Active Directory aus 3 Schichten, die der Sicherung der Datenbankdatei *NTDS.DIT* dienen, welche die eigentliche Datenablage des Active Directorys darstellt. Den direkten Zugriff auf die Datenbank hat ausschließlich die ESE (Extensible Storage Engine). Über dieser liegt die Datenbankschicht, die als Puffer und Abschirmung des Direktzugriffs dient. Die Schnittstelle zur Anwendung bildet dann der im Benutzermodus laufende DSA

(Directory System Agent), der 4 verschiedene Zugriffsmechanismen bereitstellt. Diese Mechanismen dienen der Kommunikation von Applikationen mit der Active Directory Datenbank. Es sind

1. LDAP,
2. REPL (Replikation),
3. MAPI (Message API) und
4. SAM (Security Accounts Manager).

Letzterer dient der Kommunikation mit WinNT-Clients und wird im gemischten Modus auch zur Replikation auf WinNT-BDCs genutzt. MAPI wird von Exchange-Applikationen genutzt, beispielsweise wird Outlook darüber der Zugriff auf Adressbuchfunktionalitäten gewährleistet. REPL ist eine proprietäre Schnittstelle, die der Kommunikation mehrerer DSAs miteinander dient, dabei explizit der Replikation von Active Directorys über Protokolle wie SMTP und IP. LDAP ist das primäre Zugriffsprotokoll für Active Directory und alle AD-Applikationen oder nativen LDAP-Clients nutzen diesen Zugang.

Auch wenn LDAP der primäre Zugang zum Active Directory ist, nennt doch [MR99] X.500 den Paten des eigentlichen Verzeichnisdienstes. So sind das Verzeichnismodell, das Verzeichnisschema und die Namenskonvention nach X.500 gestaltet worden. Die Notwendigkeit der Konzentration auf Internetprotokolle hat aber die Wahl des Zugriffsprotokolls auf LDAP fallen lassen. Allerdings läßt auch X.500 als Vorbild der SLAP-Server diesen Umstand der X.500-Nähe erklären.

Hinsichtlich der Namenskonventionen bietet Active Directory neben der Vorgaben nach X.500 mit Distinguished Names weitere Namensformate, um Objekte im Verzeichnis zu bestimmen. So nennt [MR99]:

- Namensformat nach RFC1779,
- das LDAP-URL-Format,
- Zugriff mittels HTTP-URL,
- das Windows-typische UNC-Format und
- das UPN-Format.

Das Format nach RFC1779 entspricht dem Attributed Naming nach X.500, wobei im Gegensatz dazu die Nennung des Pfades durch den DIT vom Blatt zur Wurzel benannt wird. Das LDAP-URL-Format ist relativ klar, da dort nach der URL-typischen Nennung des Protokolls der Servername gefolgt wird vom DN des Objekts. Beim HTTP-URL-Format ist es ähnlich, nur dass die Trennung der Stufen im DIT durch Slashes geschieht. Das bekannte Format des UNC's ist der Windows-typische Aufbau mittels Backslashes, die dann URL-ähnlich strukturiert sind. Das letztgenannte Format des User Principal Name (UPN) ist sehr stark an das bekannte Format nach RFC822 angelehnt, welches bei SMTP E-Mail genutzt wird. Ein Beispiel ist dabei *Nutzer@Domäne.de*.

Alle diese Formate werden zeitgleich unterstützt und mittels des DSA interpretiert. Dadurch wird die Abwärtskompatibilität, aber auch die Neuausrichtung an offenen Standards gewährleistet.

4.1 Struktur einer Active Directory Domäne

Eine Domäne besteht in ihrer kleinsten Form aus einem Domaincontroller und angeschlossenen Windows- in manchen Fällen auch Unix-Clients. Der Domaincontroller stellt dabei als grundsätzlichen Dienst das Active Directory zur Verfügung, welches als Datenablage aller Informationen rund um die Domäne dient. Dieser Controller ist der zentrale Punkt der Authentifizierung und die Informationsquelle zu allen in der Domäne angebotenen Ressourcen. Als weiterer wesentlicher Aspekt der Neuerungen, die mit Active Directory eingeführt wurden, ist zu erwähnen, dass keine Unterscheidung mehr in Primary (PDC) und Backup Domaincontroller (BDC) stattfindet. Vielmehr gibt es eine Multimaster-Struktur und für spezielle Aspekte Betriebsmasterrollen. Vertiefend sei neben der bereits genannten Literatur auf 4.4 verwiesen. Ebenso unterstützt Active Directory so genannte verschiedene Funktionsebenen, die festlegen, welche Features nutzbar sind und welche Betriebssysteme als Domaincontroller eingesetzt werden können. Näheres kurz in 4.5.

Da der DNS-Dienst zur Auffindung dieser Informationsquelle genutzt wird, muss natürlich in jeder Domäne ein funktionierender DNS-Server erreichbar sein. Die vorgesehene Standardstruktur einer Windows-Domäne bietet daher an, den Windows-DNS-Server ebenfalls auf dem Domaincontroller zu installieren. Die Anwendung des broadcastlastigen NetBIOS-Protokolls wird nur noch via TCP/IP unterstützt, zumal Rechner einer Domain nicht in einem Netzsegment lokalisiert sein müssen.

Verschiedene Einsatzszenarien machen den Einsatz von mehreren Servern notwendig oder sinnvoll. Als ein erstes fällt einem da die Strukturierung in mehrere Domänen innerhalb eines Baumes ein, wobei jede Domain auf einem eigenen Controller gehostet wird. Aber auch innerhalb einer Domain ist der Einsatz mehrerer Controller denkbar, so etwa zum Zwecke der Performance-Steigerung bei vielen Clients, zur Steigerung der Ausfallsicherheit des Dienstes, in dem ein Server als Backup des anderen wirkt, aber auch wenn die Organisation, um deren Domäne es geht, sich physikalisch über mehrere Standorte erstreckt, deren Leitungsanbindung untereinander nicht leistungsstark ist und daher den Einsatz eines Controllers je Standort sinnvoll macht. Neben der angesprochenen Strukturierung mittels Subdomänen, in deren Folge ein Domänenbaum entsteht, gibt es im Windows-Umfeld auch die Zusammenfassung mehrerer parallel bestehender Domänenbäume in einem Domänenwald (Domain-Forrest). Diese Menge aller zusammengefasster Domänenstrukturen nennt sich dann Gesamtstruktur.

Es ist zu beachten, dass sich organisatorische und geografische Strukturierungsbegriffe im AD-Umfeld ein wenig durchmischen und wie im realen Leben auch, keine einheitliche Hierarchie bilden. So gibt es in der organisatorischen Struktur die Domäne, die Subdomäne, welche als solches nicht existiert, sondern eine untergeordnete Domäne in einem Domänenbaum bildet, und die Organisationseinheiten innerhalb einer Domäne, welche diese wiederum beliebig durch paralleles und verschachteltes Vorkommen strukturieren. Diese Elemente dienen der Abbildung einer Unternehmensstruktur in das System. Wie so eine OU innerhalb einer Domäne aussehen kann, zeigt Abbildung 6.3.

Der eingeworfene geografische Strukturbegriff ist der des Standortes, welcher eine ortsabhängige Strukturierung innerhalb einer Domäne, aber auch domänenübergreifend schafft. Das bedeutet, dass an einem Standort sowohl mehrere Domänen realisiert sein können, wie

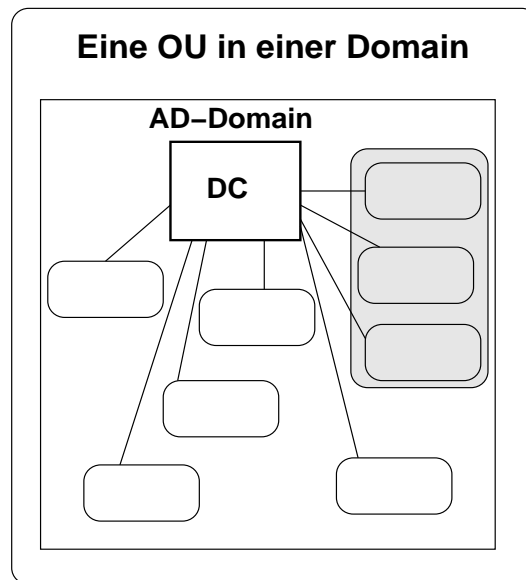


Abbildung 4.1: Eine Organisationseinheit in einer Domäne

Abbildung 4.2 zeigt, aber auch eine Domäne sich über mehrere Standorte erstrecken kann, wie es in Abbildung 4.3 der Fall ist.

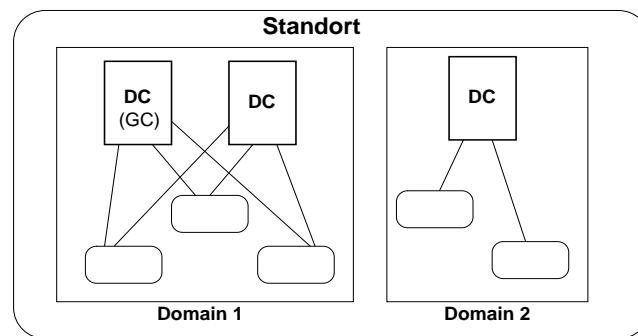


Abbildung 4.2: Standort mit 2 Domänen

Im Endeffekt ist zu beachten, dass jeder Computer einer Domäne und einem Standort zugeordnet ist, was Auswirkungen auf seine administrativen Vorlagen und sein Verhalten bei Authentifizierung und Ressourcensuche haben kann.

Grundsätzlich kann es auch eine Durchmischung der Einzelfälle geben, wie es Abbildung 4.4 zu sehen ist. Es ist auch möglich, wie in Abbildung 4.5, dass sich eine OU innerhalb einer Domäne über mehrere Standorte erstreckt.

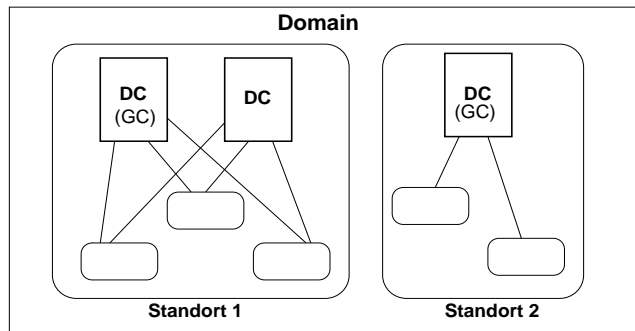


Abbildung 4.3: Domäne an 2 Standorten

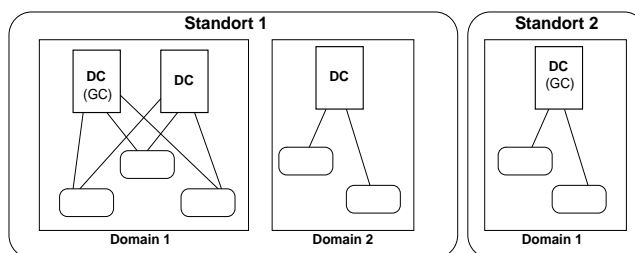


Abbildung 4.4: Mehrere Standorte und Domänen

4.2 Der globale Katalog

Neu im Active Directory ist ein Suchindex, der die wichtigsten Suchattribute des Verzeichnisses indiziert und damit einen schnelleren Zugriff oder eine unscharfe Suche auf Objekte gestattet. Dieser Index enthält unter anderem die Informationen, die eine zügige, ab der Funktionsebene „Windows 2000 pur“ überhaupt eine Authentifizierung und Anmeldung an der Domäne ermöglichen. Im gemischten Betriebsmodus ist diese auch noch über die NTLM-Authentifizierung mittels des PDC-Emulators möglich. Außerdem wird der Netzverkehr innerhalb der Domäne dank der Suchfunktion deutlich verringert, was bis dahin durch NetBIOS-Broadcasts erfolgte.

Begrifflich wurde der Index unter dem Namen „Globaler Katalog“ eingeführt und auf dem ersten Domaincontroller der Stammdomäne eines Domänenbaums bereitgestellt sowie durch Teilreplikation des Verzeichnisses aktualisiert wird.

Sämtliche Suchen über Daten im AD werden über den globalen Katalog durchgeführt, welcher sämtliche Objekte des Active Directory enthält. Zur Datenminderung und damit Beschleunigung der Suche werden aber nur die wichtigsten Attribute aus dem Verzeichnis in den GC repliziert. Festgelegt wird die Auswahl durch das Gesamtstruktur-Schema. Der globale Katalog enthält Informationen über alle Domänen einer Gesamtstruktur, was zu einer entsprechenden Größe des Katalogs führen kann. Als Richtgröße nennt [Tie03], dass etwa 40 bis 50 % der Daten im AD einer Domäne in den globalen Katalog der Gesamtstruktur repliziert werden. Dies ist bei der Planung der Speicherkapazität der Domaincontroller zu bedenken, die als GC-Server dienen.

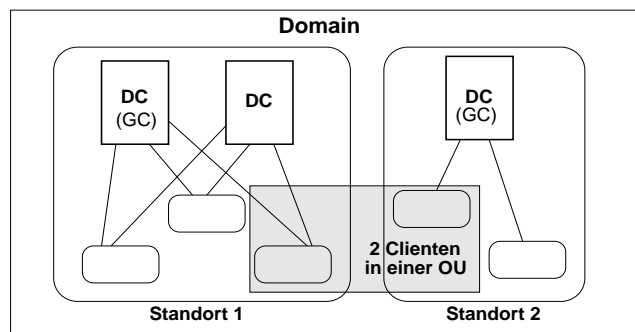


Abbildung 4.5: Eine Organisationseinheit in einer Domäne über mehrere Standorte erstreckt

Da konzeptionell jede Form der Suche auf dem globalen Katalog ausgeführt wird, ist auch der Einsatz eines Active Directory als LDAP-Ersatz nur via des GC-Servers vorgesehen. Anonyme Zugriffe auf Daten des Verzeichnisses werden ausschließlich auf den GC zugelassen. Authentifizierte Nutzer können sich natürlich auch direkt an einen Domaincontroller binden, werden aber dort nur die Daten der jeweiligen Domäne vorfinden und können sich nur mittels domainlokaler Nutzer anmelden.

Auf dem ersten Domaincontroller einer Gesamtstruktur wird automatisch der globale Katalog der Gesamtstruktur eingerichtet. Dies kann in der Folge manuell beeinflusst werden, sollte aber nach Aussagen von [KT04] und [Tie03] wohl bedacht geschehen. Der globale Katalog ist ein wesentliches Werkzeug im Rahmen der Authentifizierung von Nutzern an der Domain. Jedes Mal wird ein Katalogserver durch den Domaincontroller, der die Objektauthentifizierung durchführt, kontaktiert. Daher ist es gegebenenfalls sinnvoll, auf mehr als einem Domaincontroller einen Katalog einzurichten. Vor allem bei Strukturierung mittels Standorten und damit verbundenen WAN-Leitungen, die hohe Kosten verursachen oder langsame Verbindungen bereitstellen, ist es sinnvoll, einen Katalogserver pro Standort anzubieten. Nur bei sehr kleinen Standorten rät [KT04], den Einsatz eines eigenen Katalogservers genau zu überdenken. Der Grund ist die daraus resultierende Datenmenge, die durch die Replikation entstehen kann. Aber auch standortintern bedeuten mehrere GC-Server einen starken Replikationsaufwand und damit verbundenen Netzverkehr. Auf der anderen Seite kann die Bereitstellung mehrerer GC-Server die Performance bei der Suche oder Authentifizierung deutlich erhöhen.

Ist der GC durch den authentifizierenden Domaincontroller nicht erreichbar, kann keinerlei Authentifizierung mehr über diesen Domaincontroller stattfinden, die Gesamtstruktur ist über diesen DC sozusagen nicht mehr erreichbar. Dies passiert nur ab der Funktionsebene Windows 2000 pur aufwärts, der Mischbetrieb mit Windows NT wird hier jedoch sowieso außen vorgelesen. Der Grund für das Scheitern liegt in der notwendigen Abfrage der Zugehörigkeit des zu authentifizierenden Objekts zu universellen Gruppen. Diese Gruppen stehen nicht nur innerhalb einer Domäne oder Domänenstruktur (Tree) zur Verfügung, sondern durch ihre Speicherung im GC gesamtstrukturweit. Der Umstand des Scheitern unter bestimmten Voraussetzungen hat in Windows 2003 zu einer Erweiterung des Funktionsumfangs geführt, der darin besteht, dass auf jedem DC ein Caching von Zugehörigkeiten zu

universellen Gruppen erfolgen kann. Dies kann in den *NTDS Site Settings* des Active Directory Standorte-SnapIns eingestellt werden, wie es Abbildung 4.6 zeigt, und es reicht, einen DC pro Standort mit Windows 2003 auszustatten, um dieses Feature am Standort zu nutzen.

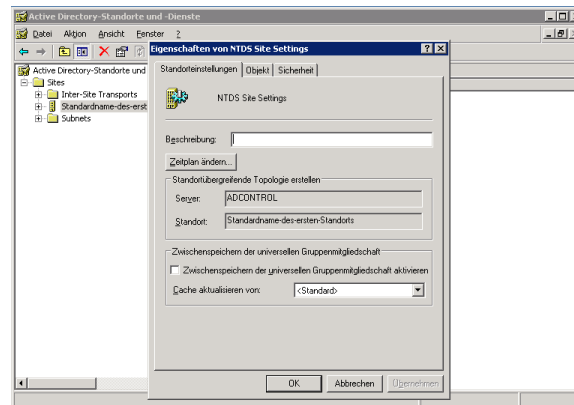


Abbildung 4.6: Zwischenspeichern der universellen Gruppen aktivieren

Neben all diesen Grundstrukturen des Active Directorys an sich, ist zu erwähnen, dass Active Directory einerseits nur durch die Bereitstellung anderer Dienste funktioniert, im Gegenzug aber auch als Datenablage vieler Dienste dient. Darauf soll im Folgenden detaillierter eingegangen werden.

4.3 DNS und AD

Active Directory nutzt seit seiner Ausrichtung auf Internet-Standards den Domain Name Service als zentrales Werkzeug zum Auffinden von Ressourcen und der Lokalisierung von Diensten.

Wie schon angerissen, wird zu diesem Zwecke vorausgesetzt, dass der DNS-Dienst so genannte SRV-Einträge Software Resource Records (SRV) unterstützt. Diese Funktionalität wird seitens des Clienten und des Servers benötigt. Alle Windows-Betriebssysteme, die mit Active Directory nativ zusammenarbeiten, unterstützen diese Funktionalität von Hause aus. Solche Einträge verbinden einen Dienstenamen innerhalb einer DNS-Domäne mit dem Namen eines Servers, dem TCP- beziehungsweise UDP-Port des Dienstes und einer Priorität des Diensteanbieters. Dadurch kann der Client mittels der DNS-Funktionalität dynamisch an wechselnde Gegebenheiten seitens der Serverlandschaft gebunden werden. Es wird also ermöglicht, ohne Änderungen der Konfiguration am Clienten, virtuelle Adressen wie bei Clustern ohne Broadcasts oder eine sinnvolle Lastverteilung durch die Server selbst zu realisieren oder bei Ausfall beziehungsweise Nichterreichbarkeit eines Servers die Backup-Variante ausfindig zu machen und anzusprechen.

Doch gerade um diese Aktualität zu gewährleisten und auf strukturelle Änderungen des Netzes zu reagieren, wird empfohlen, darüber hinaus einen Server einzusetzen, der dynamische Updates erlaubt. So können Windows-Clients bei Adressvergabe mittels DHCP und dynamischen Adressbereichen ihre jeweilige Konfiguration aktuell an den Server vermitteln, aber

auch die gegenseitige Überwachung zwischen den Domaincontrollern die Belastung oder Erreichbarkeit untereinander beeinflussen.

Für jede Domain werden viele verschiedene Einträge im DNS erstellt, die zum Auffinden des Verzeichnisses, des globalen Katalogs und des Authentifizierungsdienstes genutzt werden. Wie bereits in der Strukturdiskussion erwähnt, gibt es die Aufspaltung einer Domäne in verschiedene Standorte beziehungsweise die Möglichkeit dazu. Dies findet sich natürlich auch in den DNS-Einträgen wieder. Gleichzeitig wird jedem Objekt eine SID zugeordnet, die sich auch in manchen Einträgen wieder findet.

Die bei einer standardmäßigen Einrichtung angelegten DNS-Einträge werden in 2 Subdomänen strukturiert und finden sich in folgender Liste:

<i>_msdcs.domain.tld.</i>					
<i>_ldap.Standortname._sites.dc</i>	SRV	0	100	389	<i>Domaincontroller</i>
<i>_kerberos.Standortname._sites.dc</i>	SRV	0	100	88	<i>Domaincontroller</i>
<i>_ldap._tcp.dc</i>	SRV	0	100	389	<i>Domaincontroller</i>
<i>_kerberos._tcp.dc</i>	SRV	0	100	88	<i>Domaincontroller</i>
<i>_ldap._tcp.DomainID.domains</i>	SRV	0	100	389	<i>Domaincontroller</i>
<i>_ldap.Standortname._sites.gc</i>	SRV	0	100	3268	<i>Domaincontroller</i>
<i>_ldap._tcp.gc</i>	SRV	0	100	3268	<i>Domaincontroller</i>
<i>_ldap._tcp.pdc</i>	SRV	0	100	389	<i>Domaincontroller</i>
<i>gc</i>	A				<i>DomaincontrollerIP</i>
<i>domain.tld.</i>					
<i>_ldap._tcp.Standortname._sites</i>	SRV	0	100	389	<i>Domaincontroller</i>
<i>_kerberos._tcp.Standortname._sites</i>	SRV	0	100	88	<i>Domaincontroller</i>
<i>_gc._tcp.Standortname._sites</i>	SRV	0	100	3268	<i>Domaincontroller</i>
<i>_ldap._tcp</i>	SRV	0	100	389	<i>Domaincontroller</i>
<i>_kerberos._tcp</i>	SRV	0	100	88	<i>Domaincontroller</i>
<i>_gc._tcp</i>	SRV	0	100	3268	<i>Domaincontroller</i>
<i>_kpasswd._tcp</i>	SRV	0	100	464	<i>Domaincontroller</i>
<i>_kerberos._udp</i>	SRV	0	100	88	<i>Domaincontroller</i>
<i>_kpasswd._udp</i>	SRV	0	100	464	<i>Domaincontroller</i>
<i>_ldap._tcp.Standortname._sites. DomainDnsZones</i>	SRV	0	100	389	<i>Domaincontroller</i>
<i>_ldap._tcp.DomainDnsZones</i>	SRV	0	100	389	<i>Domaincontroller</i>
<i>DomainDnsZones</i>	A				<i>DomaincontrollerIP</i>
<i>_ldap._tcp.Standortname._sites. ForestDnsZones</i>	SRV	0	100	389	<i>Domaincontroller</i>
<i>_ldap._tcp.ForestDnsZones</i>	SRV	0	100	389	<i>Domaincontroller</i>
<i>ForestDnsZones</i>	A				<i>DomaincontrollerIP</i>
<i>Domaincontroller</i>					<i>DomaincontrollerIP</i>

An die entsprechenden Einträge der Tabelle ist die übergeordnete DNS-Domäne anzuhängen, um den jeweiligen Eintrag zu erhalten.

Die Einträge der Tabelle werden für jeden Domaincontroller angelegt, einzig die Einträge des PDC und des GC sind auf eine Auswahl an Domaincontrollern beschränkt. Dadurch

entsteht eine ziemlich lange Liste, die sich wahrlich mit statischem DNS auf Grund ihrer Unübersichtlichkeit und der Durchmischung mit Werten wie Domainen-IDs schlecht pflegen lässt.

Aber die Liste enthält auch große Mengen redundanter Daten, die erst in einem sehr stark strukturierten Ausbau von Active Directory Domänen Auswirkungen zeigen beziehungsweise ein paar DNS-Requests weniger verursachen. Solange sich eine Domain beispielsweise nicht über mehrere Standorte erstreckt, also keinerlei Unterschiede für den Clienten entstehen, welchen Domaincontroller er auch kontaktiert, solange sind auch die Einträge mit *Standortname*-Bestandteil vernachlässigbar und kürzen die Tabelle deutlich. Auch die typischen Windows-Einträge, die auf die DomainID aufsetzen, dienen zwar der internen Performance-Steigerung, können aber unter bestimmten Gesichtspunkten auch aus der Liste gestrichen werden. Wie bereits angesprochen, ist die DomainID die SID des Domain-Objektes im AD-Verzeichnis. Diese ist einmalig und eindeutig und ändert sich auch, wenn das Objekt gelöscht und mit selben Namen neu angelegt wird. Somit ist die DomainID-basierte DNS-Suche eindeutiger und wird von Clienten als erste versucht. Schlägt diese Suche nach der DomainID im DNS allerdings fehl, wird automatisch auf die namensbasierte Suche zurückgegriffen und der Zugriff auf AD-basierte Ressourcen nur um einen DNS-Request verzögert. Probleme oder Uneindeutigkeiten können natürlich nur entstehen, wenn eine Domäne gelöscht und eine Neue mit gleichem Namen angelegt wird, auf die dann mit den alten DNS-Einträgen verwiesen wird. Da aber auch alle Objekte der Domain eine neue SID haben, was sich aus dem Aufbau der IDs erklärt¹, wird nur dann im folgenden Schritt der Zugriff abgewiesen, da auch die Berechtigungen an SIDs gebunden sind.

Im Falle der Einbindung einer Subdomäne „sub“ in den Domänenbaum von „domain.tld“ werden folgende DNS-Einträge ergänzend für jede dieser Subdomänen angelegt:

_msdcs.domain.tld.

<i>_ldap._tcp.SubdomainenID.domains</i>	SRV	0	100	389	<i>Subdomaincontroller</i>
---	-----	---	-----	-----	----------------------------

_msdcs.sub.domain.tld.

<i>_ldap._tcp.Standortname._sites</i>	SRV	0	100	389	<i>Subdomaincontroller</i>
<i>_kerberos._tcp.Standortname._sites</i>	SRV	0	100	88	<i>Subdomaincontroller</i>
<i>_ldap._tcp.pdc</i>	SRV	0	100	389	<i>Subdomaincontroller</i>
<i>_ldap._tcp.dc</i>	SRV	0	100	389	<i>Subdomaincontroller</i>
<i>_kerberos._tcp.dc</i>	SRV	0	100	88	<i>Subdomaincontroller</i>

sub.domain.tld.

<i>_ldap._tcp</i>	SRV	0	100	389	<i>Subdomaincontroller</i>
<i>_kerberos._tcp</i>	SRV	0	100	88	<i>Subdomaincontroller</i>
<i>_kpasswd._tcp</i>	SRV	0	100	464	<i>Subdomaincontroller</i>
<i>_kerberos._udp</i>	SRV	0	100	88	<i>Subdomaincontroller</i>
<i>_kpasswd._udp</i>	SRV	0	100	464	<i>Subdomaincontroller</i>
<i>_ldap._tcp.Standortname._sites</i>	SRV	0	100	389	<i>Subdomaincontroller</i>
<i>_kerberos._tcp.Standortname._sites</i>	SRV	0	100	389	<i>Subdomaincontroller</i>
<i>Subdomaincontroller</i>	A				<i>SubdomaincontrollerIP</i>

¹siehe auch Betriebsmasterrollen in der Folge

Auch hierbei sind die Redundanzen zu beachten, deren Notwendigkeit nicht in allen Fällen gegeben ist. In dieser Liste ist wieder ein Domaincontroller beispielhaft angegeben, bei mehreren finden sich entsprechende Einträge für jeden DC. Einzig der Eintrag des PDC-Emulators, der mit der Betriebsrolle korrespondiert und damit einmalig in der Domain ist, kann und darf nur einmalig vorhanden sein. Weiterhin ist zu bedenken, dass ein globaler Katalog immer für die Gesamtstruktur gültig ist, daher auch ein Domaincontroller einer Substruktur, der als GC-Server dient, einen DNS-Eintrag für den Katalog im DNS-Baum der Root-Domain erhält.

Welche Auswahl dieser DNS-Einträge für den Betrieb wirklich notwendig sind, wird im Kapitel der Lösungsideen unter 5.5 diskutiert. Wie erwähnt, dienen die Redundanzen nur bei starker Strukturierung und auch da vor allem der Performance-Steigerung und Lastverteilung und Minderung des Netzverkehrs.

4.4 Replikation, Betriebsmasterrollen und Backup

Einen wesentlichen Unterschied zur Begriffswelt von Windows NT ergibt sich noch aus der notwendigen Replikation der Daten der Domäne von einem Controller zum anderen. Bei früheren Windows-Versionen gab es einen Primary Domaincontroller (PDC) und beliebig viele Backup Domaincontroller (BDC). Einzig auf den PDC konnte schreibend zugegriffen werden, um Informationen zur Domain zu ändern. Die BDC dienten nur als Read-Only-Kopie der Datenbasis. Dieses Konzept, welches als Single-Master-Replikation bekannt ist, wurde mit Einführung des Active Directorys aufgegeben und durch ein Multi-Master-System ersetzt. Dadurch können Veränderungen von Daten im Active Directory nun auf jedem Domaincontroller durchgeführt werden, die in der Folge durch eine Replikation auf alle Domaincontroller der Domäne und zu Teilen in den GC repliziert werden. Die Replikation erfolgt durch den Abgleich von inkrementellen Versionsnummern, die den Objekten des Verzeichnisses zugeordnet werden, auf den verschiedenen Domaincontrollern, wobei die jeweils letzten Änderungen dann über alle Controller der Domäne verteilt werden. Diese inkrementellen Versionsnummern werden automatisch durch den KCC (Knowlegde Consistency Checker) verwaltet und für den Vergleich hält jeder DC einer Domäne die letzten ihm bekannten Versionsnummern aller restlichen DCs in einer Liste. Beim globalen Katalog erfolgt die Replikation anhand der veränderten Attribute, welche für eine Replikation in den GC laut Schema vorgesehen sind. Dabei wird unter Windows 2000 jeweils der gesamte Katalog repliziert, wenn es auch nur eine einzige Attributänderung gab. Unter Windows 2003 wurde diese ungünstige Lösung dahingehend verbessert, dass nur noch Änderungen repliziert werden. Außerdem wird diese Multi-Master-Replikation ausschließlich zwischen Domaincontrollern mit Windows 2000 und 2003 durchgeführt. Sollten in der Funktionsebene „Windows 2000 gemischt“ oder „Windows Server 2003 - interim“ noch WindowsNT-BDC betrieben werden, wird deren Replikation mittels eines PDC-Emulators im Single-Master-Prinzip durchgeführt.

Neben diesen Anmerkungen zur Replikation sei auf das Konzept der Betriebsmaster eingegangen. Diese Master sind notwendig, da es auch weiterhin Aktionen gibt, die auch in einem Multi-Master-System einer atomaren Ausführung bedürfen. Solche Aktionen werden dann auf dem jeweiligen Betriebsmaster durchgeführt. Im Englischen wird der Begriff für

solche Aktionen mit FSMO (Flexible Single Master Operations) abgekürzt, auch wenn die Flexibilität in der grundsätzlichen Übertragbarkeit der Rollen besteht, was allerdings manuell geschehen muss. Es gibt 5 Betriebsmasterrollen, wobei zwei für die Gesamtstruktur gelten und 3 für die jeweilige Domain. Jede dieser Rollen darf zeitgleich jeweils nur ein einzelner DC innerhalb der Gesamtstruktur beziehungsweise der Domäne inne haben.

Für die Gesamtstruktur gibt es

- den *Schemamaster*, welcher Änderungen am Active Directory Schema überwacht und
- den *Domänennamen-Master*, der das Hinzufügen und Entfernen von Domänen in der Gesamtstruktur übernimmt.

Um diese Betriebsmaster-Rollen einem anderen Domaincontroller der Gesamtstruktur zu übertragen, was im Falle von längerfristigen Wartungsarbeiten sinnvoll sein kann, muss der Administrator dies auf dem Zielcontroller im MMC-SnapIn *Active Directory Schema* beziehungsweise *Active Directory Domänen und Vertrauensstellungen* anstoßen.

Demgegenüber stehen die drei domänenweiten Betriebsmaster-Rollen

- des RID-Masters, der für die Zuteilung von SID-Blöcken an die Domaincontroller der Domäne verantwortlich ist,
- der PDC-Emulation, welcher der Replikation auf WindowsNT-BDCs und der Authentifizierung von Nutzern an Windows 95/98/ME/NT-Clients dient und
- des Infrastrukturmasters, der alle Änderungen überwacht, die domänenex- und -interne Objekte verbinden.

Die letztgenannte Rolle dient also der Anpassung von Objekt-Verweisen, die domänenübergreifend wirksam werden. Ist etwa ein Domänen-Nutzer Mitglied einer lokalen Gruppe einer fremden Domain, an der Anpassungen vorgenommen werden, aktualisiert der Infrastrukturmater die Verweise am beziehungsweise zum lokalen Nutzer-Objekt.

Bezüglich der Rollen des Domänennamen- und RID-Masters soll hier noch kurz auf den Aufbau von SIDs und den Einfluss der beiden Rechner auf diesen eingegangen werden. Eine SID ist eine eindeutige Zahl, deren eine Hälfte aus der DomainID besteht und die zweite Hälfte als innerhalb der Domain eindeutige Zahl durch den RID-Master festgelegt wird. Die DomainID wird natürlich bei deren Einrichtung festgelegt und durch den Domainnamemaster als innerhalb der Gesamtstruktur eindeutige Zahl generiert. Wie sonst auch sei für weiter vertiefende Information auf die vielseitige Literatur rund um das Thema verwiesen.

Neben der durch den KCC gesteuerten und automatisch erfolgenden Replikation der Daten zwischen den Domaincontrollern einer Domäne ist das Backup des Active Directorys im Rahmen seines Einsatzes als Dienst eine wesentliche Komponente zur Sicherung des dauerhaften Betriebs. Prinzipiell kann hierzu der Windows Backup Dienst genutzt werden, hat aber gegenüber Produkten von Drittanbietern wohl seine Mängel. Ein Sichern der AD-Datenbankdatei oder des SYSVOL-Ordners ist nicht ausreichend, da das Active Directory nur durch Rekonstruktion des Registrystandes und sämtlicher Systemumgebungeinstellungen funktional wieder hergestellt werden kann. Weiterhin wird bei [KT04] auch auf den Umstand hingewiesen, dass eine Wiederherstellung einer AD-Sicherung nur innerhalb des Zeitraumes von 60 Tagen möglich ist, da dies dem Zeitfenster von zum Löschen markierten

Objekten entspricht, wodurch Inkonsistenzen durch Löschungen entgegen gewirkt werden soll. Außerdem muss bei Windows 2000 mindestens das Service Pack 2 eingespielt sein, da mit diesem Fehler in der Backup-API beseitigt wurden. Nach [Bod05] sei aber darauf hingewiesen, dass im regulären Betrieb das Zurückstellen einer Sicherung nur im absoluten Ausnahmefall notwendig sein dürfte. Fällt ein Domaincontroller aus, wird er neu eingerichtet, erneut in die Domäne eingebunden und der automatische Replikationsmechanismus bringt die Daten des ADs auf dem Domaincontroller auf den neuesten Stand. Interessanter ist hingegen das Zurückstellen versehentlich gelöschter Objekte aus Backups. Dort ist es zwar prinzipiell denkbar, das Objekt wiederherzustellen, allerdings werden dabei auch die Zeitstempel des Backupzeitpunkts wiederhergestellt und die Replikation wird das Restore auf Grund der aktuelleren Löschung wieder überschreiben. Für solche Zwecke empfiehlt [Bod05] beispielhaft den „Aelita Recovery Manager“, inzwischen im Produktsortiment der Firma „Quest“ (siehe [Que05]). Dieser kann sowohl auf einem Server aber auch einer Administratorenworkstation laufen, wenn diese möglichst immer verfügbar ist, und stellt beispielsweise die Möglichkeit bereit, einzelne Objekte wiederherzustellen. Dabei besteht auch die Möglichkeit die Wiederherstellung mit dem aktuellen Zeitstempel zu versehen, wodurch diese auch vom Replikationsmechanismus anerkannt wird. Da es im Umfang dieser Arbeit nicht sinnvoll möglich erscheint, eine vollständige Backup-Strategie zu untersuchen, sei hier nur auf den Umstand hingewiesen und für eine Vertiefung auf entsprechende Literatur verwiesen, die auch in größerem Umfang im Literaturverzeichnis der Arbeit zu finden ist.

4.5 Funktionsebenen

Jede Domain und auch die Gesamtstruktur hat als eine Eigenschaft eine so genannte Funktionsebene. Diese bestimmt, welche Betriebssysteme jeweils auf den Domaincontrollern abwärtskompatibel unterstützt werden, ermöglichen auf der anderen Seite aber die Aktivierung bestimmter verbesserter Features des jeweiligen Betriebssystems. Dabei bestimmt innerhalb der Domain der DC mit dem ältesten Betriebssystem die maximal einstellbare Funktionsebene, beziehungsweise bei der Gesamtstruktur die Domain mit der geringsten Funktionsebene. Standardmäßig erfolgt die Einrichtung im „Windows 2000 gemischt“-Modus, welcher auch noch den Betrieb von WindowsNT-basierten BDC ermöglicht. Dazu dient der PDC-Emulationsmaster der Domain. Zwischen den WindowsNT-BDCs und dem PDC-Emulator erfolgt eine Single-Master-Replikation.

Die nächsthöhere Stufe ist der „Windows 2000 pur“-Modus, der dann die neu eingeführten „lokalen Domänengruppen“ und die „universellen Gruppen“ sowie die beliebige Verschachtelung von Gruppen ermöglicht. Die „lokalen Domänengruppen“ können Mitglieder aus der Gesamtstruktur haben, erlauben aber nur die Zugriffsberechtigung auf Ressourcen der eigenen Domain, was dem Administrator die Möglichkeit eröffnet, die Ressourcen seiner Domäne geschickt zu verwalten. Außerdem wird die Funktionalität der SID-History eingerichtet. Damit lassen sich Änderungen an Objekten, die zur Neuvergabe einer SID führten, nachvollziehen - beispielsweise bei der Umbenennung einer ganzen Domäne oder deren Verschiebung im Baum. In der Folge können Berechtigungen, die an die alte SID geknüpft waren, weiterhin auf das Objekt angewandt werden.

Die neueste Funktionsebene nennt sich „Windows Server 2003“ und bietet weitere Features, wie das Umbenennen ganzer Domänen. Allerdings setzt dies den ausschließlichen Einsatz

von Windows 2003 Servern in der Domain voraus.

Ist der Einsatz von WindowsNT-BDCs neben ansonsten ausschließlich Windows Server 2003 Domaincontrollern angedacht, gibt es noch eine Ebene „Windows Server 2003 - interim“. Diese dient dem direkten Übergang von WindowsNT auf Windows 2003. Ansonsten ist sie in der Bedeutung nebenrangig.

4.6 Authentifizierung

Das zentrale Werkzeug zur Authentifizierung ist mit Einführung von Active Directory von der lokalen NTLM-Authentifizierung auf Kerberos umgestellt worden. Dies bietet mehrere Vorteile, die durch die proprietäre Lösung nicht abgedeckt wurden. So ist einerseits die Koexistenz mit anderen Betriebssystemen erleichtert, meist sind natürlich andere Systeme als Nutzer von Windows-Ressourcen vorgesehen. Auf der anderen Seite wird dadurch die Einrichtung der Vertrauensstellungen unter Nutzung von einheitlichen einmaligen Ressourcen-Objekten ermöglicht. Eine NTLM-Vertrauensstellung sah vor, dass der Ressourcenserver beim Domaincontroller des Klienten dessen Authentifizierung nachfragte. Jetzt ist der Client beziehungsweise dessen Domaincontroller mit der Aufgabe betraut, die Berechtigungen in Form von Tickets bei den DCs der Ressourcenserver einzuholen.

Die Kerberos-Implementierung hat natürlich einige Einschränkungen oder Eigenheiten. So wird standardmäßig eine ARC4-Verschlüsselung genutzt, deren Implementierung aber beispielweise mit der Heimdal-Implementierung nicht funktioniert. Weiterhin kann eine DES-Verschlüsselung genutzt werden, was aber für Objekte im AD explizit anzugeben ist. Die Auswahl von Hashing-Algorithmen, um kryptografische Nachrichten zu signieren, ist ebenfalls mit CRC und dem standardmäßig genutzten MD5 eingeschränkt. MD4 wurde ebenso bei der Auswahl außen vor gelassen, wie das deutlich sicherere SHA1. Ist es also vorgesehen, Nicht-Windows-Clienten an ein Active Directory zu binden, wird empfohlen, explizit auf DES-Schlüsselgenerierung umzustellen. Im Umkehrschluss muss bei gemischten Umgebungen mit externen Kerberos-Systemen auf eine Verteilung von funktionsfähigen Schlüsseln in den genannten Typen geachtet werden.

Neben diesen Punkten der Kerberos-Authentifizierung gibt es einige Situationen, in denen weiterhin mittels NTLM (NT Lan Manager) authentifiziert wird. So wird zum Beispiel der Zugriff auf Netzwerkfreigaben in Form des FQDN beziehungsweise UNC mittels Kerberos gesteuert, bei Zugriff mittels der IP entscheidet jedoch immer noch der NTLM über die Berechtigungen. Ähnliche Erfahrungen berichtet die Universität von Washington auf [Was05], berichten aber auch, dass es keine abschließende Aufzählung der betreffenden Dienste gibt.

4.7 Gruppenrichtlinie

Ein Werkzeug, welches bereits aus dem WindowsNT-Umfeld² bekannt ist und im AD-Kontext weiter ausgebaut wurde, ist die Gruppenrichtlinie. Gruppenrichtlinien dienen der Steuerung von Konfigurationen von Benutzern und Computern und können auf Ebene von Standorten, Domänen oder Organisationseinheiten Anwendung finden. Mit ihrer Hilfe werden Vorgaben der Unternehmenspolitik auf Benutzer angewandt, der Zugriff auf Ressourcen, die

²dort als Systemrichtlinie benannt

für den Benutzer in der Domäne verfügbar sind, gesteuert und Software beziehungsweise Startmenü-Einträge dem Benutzer zugewiesen oder vor ihm verborgen. Nicht zuletzt Einträge in der Registry werden auf Domänencomputern durch Gruppenrichtlinien festgelegt oder angepasst. Auf jeder Ebene, die oben genannt wurde, können mehrere Gruppenrichtlinienobjekte definiert werden, die dann Anwendung finden.

Im Zusammenhang mit Gruppenrichtlinien fallen 3 Begriffe, die nur kurz Erwähnung finden sollen: Das GPO, der GPC und das GPT. GPO steht für Group Policy Object, also ein Objekt im AD, welches eine Gruppenrichtlinie enthält. Versionsinformationen, der Aktivitätsstatus und Strukturinformationen eines GPOs sind im zugehörigen GPC, dem Group Policy Container, aufbewahrt. Die Informationen, die die im GPC beschriebene Struktur mit Daten füllen, befinden sich im Group Policy Template.

Gruppenrichtlinien können und werden verschachtelt. Somit stellt sich die Frage, in welcher Reihenfolge GPOs abgearbeitet werden, welche Regelungen also letztendlich Anwendung finden – vor allem, bei sich widersprechenden Anweisungen unterschiedlicher GPOs. Grundlegend herrscht folgende Hierarchie zwischen den Richtlinien der genannten Ebenen:

1. Lokales GPO des einzelnen Rechners
2. GPO des Standortes
3. GPO der Domäne
4. GPO der Organisationseinheit

Gibt es innerhalb der Organisationseinheit weitere Strukturierungen in OUs werden die GPOs der OUs sozusagen spezialisierend der Struktur folgend angewandt.

Wie bereits erwähnt, kann die Definition mehrerer GPOs für ein Objekt, beispielsweise durch Zuweisung eines GPOs zum Standort und eines zur Domäne des Objektes, zu Konflikten oder widersprüchlichen Einstellungen führen. Zur Vorhersage solcher Konflikte oder um das Ergebnis der gemachten Einstellungen zu kontrollieren, gibt es ein MMC-SnapIn „Richtlinienergebnissatz“ beziehungsweise „Resultant Set of Policy“ (kurz RSoP) unter Windows XP und 2003. In dessen Protokollierungsmodus wird für ein ausgewähltes Benutzer- oder Computerobjekt die Anwendung der eingerichteten GPOs simuliert. Somit lassen sich die Auswirkungen von einer Vielzahl verschachtelter GPOs auf einen Benutzer oder Computer nachvollziehen, aber es sei auch zu Bedenken gegeben, dass jedes GPO bei der Anmeldung des Nutzers oder beim Systemstart abzuarbeiten ist, was zu einer Verzögerung selbigen führt. Aber Richtlinien bieten eine einfache und meist konsistente Art der zentralen Administration, stellen jedoch dank der Hierarchie von GPOs auch die Möglichkeit bereit, administrative Aufgaben zu delegieren.

In die Details der Möglichkeiten, die Gruppenrichtlinien bereitstellen, wird an dieser Stelle nicht eingegangen, da dies den Rahmen der Arbeit sprengen würde und es wird dazu erneut auf die umfangreiche Literatur zum Thema verwiesen. Jedoch ein interessanter Aspekt des Einsatzes soll kurz angesprochen werden. Mit Hilfe von Gruppenrichtlinien ist es möglich, Software auf beliebigen Systemen der Gesamtstruktur zu installieren, zu warten und zu aktualisieren. Dazu wird der Gruppe von Benutzern oder Computern, auf die das GPO angewendet wird, ein Paket mittels Microsoft Installer während des Systemstarts beziehungsweise der Nutzeranmeldung auf dem Zielrechner installiert und bereitgestellt. Außerdem ist

es ebenfalls möglich, dem Nutzer Software nur zu veröffentlichen. Während bei der Zuweisung von Software³ diese sofort und ohne Rückfrage auf das System installiert wird, hat bei der Veröffentlichung der Nutzer die Wahlmöglichkeit. Die Bereitstellung des Paketes wird dem Anwender signalisiert und bei Bedarf kann er dieses über das Software-Element der Systemsteuerung installieren und auch deinstallieren. Eine weitere Möglichkeit ist das Veröffentlichen mit gleichzeitigem Verbergen des Paketes in der Systemsteuerung, so dass der Nutzer dieses nicht eigenständig installieren kann. Zu einer sinnvollen Installation des Paketes kommt es in diesem Falle nur, über die dem Paket zugeordneten Dateierweiterungen, was im GPO ebenfalls festgelegt werden kann.

Nach gleichem Muster ist eine Deinstallation oder Unterbindung weiterer Neuinstallationen von Paketen einstellbar.

Voraussetzung einer Installation mittels GPO ist die Verfügbarkeit eines msi- oder msp-Paketes für die zu installierende Software. Im Rahmen der Einrichtung ist es auch möglich, einem msi-Paket verschiedene mst-Pakete zuzuordnen, die dann ebenfalls Anwendung finden. Dabei ist zu beachten, dass dies umgehend bei der Einrichtung des Paketes im GPO in den erweiterten Einstellungen zu erfolgen hat, da bei Abschluss der Einbindung sofort eine Veröffentlichung des GPOs stattfindet und das mst-File nicht mehr wirksam werden kann. Hierbei ist bereits ersichtlich, dass für den Einsatz der Installation via GPO durch den Hersteller der Software ein msi-Paket bereitgestellt werden muss oder eine Paketierung mit Hilfe externer Software notwendig ist. Eine weitere Voraussetzung ist die Bereitstellung des Paketes auf einer Netzfreigabe, die von allen betroffenen Objekten lesbar sein muss. Standardmäßig wird dazu im AD ein freigegebener Ordner erstellt, der dann die zu installierenden Pakete enthält. Alternativ ist auch die Bereitstellung über einen anderen Netzwerkpfad realisierbar, was zu einer Frage mehr bei der Einrichtung des GPOs führt. Somit sollte auch eine Bereitstellung via Samba möglich sein.

4.8 Anmerkungen zu Hardwarevoraussetzung eines Domaincontrollers

Grundsätzlich gibt es an dieser Stelle zwei Hinweise, die im Zusammenhang mit dem Betrieb eines Domaincontrollers und der damit verbundenen Bereitstellung eines Active Directorys stehen.

Eine einfache Bemerkung, die sich aus den Anforderungen des KCCs ergibt, stellt fest, dass ein Domaincontroller über zwei Netzwerkverbindungen verfügen soll. Über diese doppelte Anbindung soll abgesichert werden, dass selbst bei Ausfall der Primäranbindung ein Datenaustausch zu anderen Domaincontrollern sichergestellt wird. Damit soll Inkonsistenzen vorgebeugt werden, die durch Änderungen auf dem durch den Netzausfall abgetrennten DC entstehen könnten. Durch die Fallback-Anbindung soll dieser noch in der Lage sein, seine Änderungen in das Active Directory zu propagieren. Andererseits will man sicherstellen, dass es nicht zum Totalausfall einer Domäne kommt, wenn die Primäranbindung eines GC-Servers oder Betriebsmasters gestört ist. Über die zweite Anbindung sollen andere Domaincontroller befähigt werden, den entsprechenden Betriebsmaster noch zu erreichen und

³bei Computerobjekten einzige Möglichkeit

damit über die anderen Domaincontroller die Domäne verfügbar zu halten.

Die zweite hier wesentlich schwieriger zu bestimmende Aussage bezieht sich auf den Bedarf von Festplattenkapazität eines Domaincontrollers. Es ist zu bedenken, dass ein Domaincontroller ein volles Abbild des Active Directorys seiner Domäne hält, wobei die Datenbankdatei und die Transaktionslogs zu betrachten sind. Im Rahmen der Teststellung war diese mit circa 10 MB recht übersichtlich, allerdings ist es schwerlich abzusehen, wie sich diese bei einer Ablage mehrerer tausend Objekte durch die Integration der Nutzerdaten entwickelt. Nach dem Anlegen von 10000 Nutzerobjekten im Rahmen der Tests durch den automatischen Skript zur Datenübernahme, war die Datei auf das zwölffache angewachsen und enthielt noch wenige bis keine Informationen von Subdomänen im globalen Katalog. Überhaupt waren universelle Gruppen kaum genutzt bis dahin, was den GC deutlich vergrößern dürfte. Denn darüberhinaus ist zu beachten, dass ein Domaincontroller, der auch als globaler Katalogserver dient, weitere Objektdaten aus allen Domänen der Gesamtstruktur in seiner Datenbank ablegt. Wie [Tie03] erwähnte, nochmals etwa 40-50% aller Objektdaten aller Domänen. Sind also weitere Domänen in der Gesamtstruktur vorhanden, die viele Informationen im Active Directory ablegen, kann die Datenmenge und damit der Platzbedarf auf einem GC-Server rapide ansteigen.

5 Lösungsideen und Alternativen

In diesem Kapitel soll erläutert werden, wie an die verschiedenen Aspekte der Dienstintegration herangegangen wurde, welche Versuche unternommen wurden, wie die zur Verfügung gestellte Testumgebung aussah und welche Ansätze eine mögliche Lösung darstellen.

5.1 Grundidee einer Gesamtlösung

In der Anfangsphase ergab sich aus den Vorgaben der Aufgabenstellung eine Vorstellung der Lösungsstruktur, die möglichst nahe an einer typischen Windowsumgebung Anlehnung findet und nur in den genannten Punkten durch externe Systeme beeinflusst wird.

Die ursprüngliche Lösungsidee bestand darin, ausschließlich seitens der Windowsserver Anpassungen vornehmen zu müssen und für die Clienten transparent die Eingliederung in die bestehende Dienststruktur zu realisieren. Mit recht geringem Aufwand des Einlesens war ersichtlich, dass prinzipiell die Forderungen an den Domain Name Service auch durch externe Systeme abgedeckt werden können. Zum Anschluss an das Kerberos-System hoffte ich, eine Konfigurationsmöglichkeit im Windows 2003 Server zu finden, welche eine generelle Weiterleitung von Kerberosanfragen oder wenigstens eine solche für dem Server unbekannte Principals realisiert. Hinsichtlich der Schnittstelle zur MoUSE war von vornherein klar, dass nur eine selbstentworfene Schnittstelle möglich ist.

Insgesamt sollte es beispielsweise wie in Abbildung 5.1 aussehen.

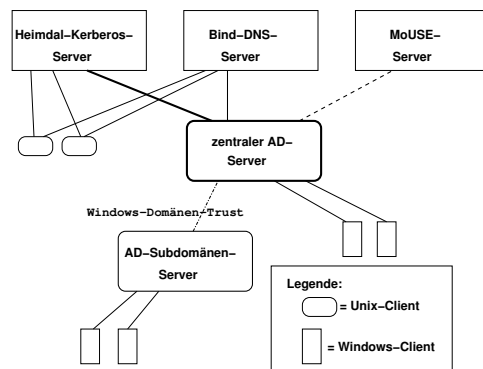


Abbildung 5.1: Eine Möglichkeit einer Gesamtlösung

5.2 Die Teststellung

Zur Erprobung meiner Ideen wurden mir 6 PCs mit 800MHz-Athlon-CPU als Teststellung zur Verfügung gestellt. 2 Stück, die als AD-Server dienen sollten, waren mit 512MB Ar-

beitsspeicher ausgerüstet, die Restlichen, welche als WindowsXP-Clients beziehungsweise Linux-Dienstserver betrieben werden sollten, waren mit 256MB bestückt. Alle Rechner hatten ca. 20GB Festplattenkapazität.

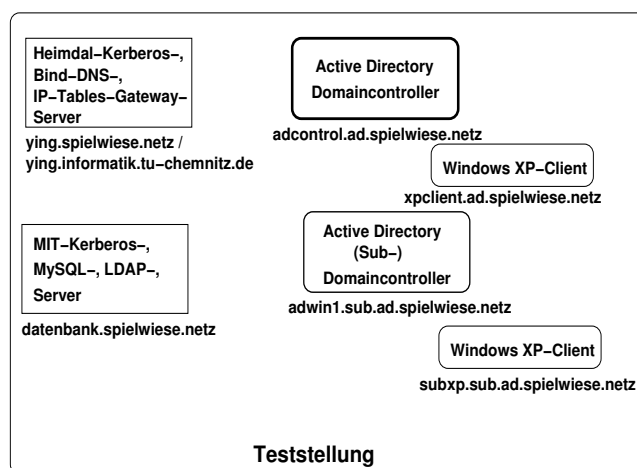


Abbildung 5.2: Struktur der Teststellung

Wie Abbildung 5.2 zeigt, war es geplant, zwei Rechner mit Linux zu betreiben, die als Simulationsumgebung der URZ-Dienststruktur dienen sollten. So wurde als erstes ein Rechner mit Linux eingerichtet, auf dem ein BIND-DNS Server und ein Heimdal Kerberos 0.6.3 installiert wurde. Weiterhin wurde auf diesem Rechner mittels *iptables* die Schnittstelle ans Campusnetz geschaffen und sichergestellt, dass keine Pakete, die nur innerhalb der Teststellung sinnvoll waren, ins Campusnetz gelangten. Auch wurde auf diesem Rechner die dynamische IP-Vergabe mittels DHCP eingerichtet. Der zweite Rechner sollte in der anfänglichen Planung als MySQL- und LDAP-Server genutzt werden, um über diesen die Datenübernahme aus den vorhandenen Quellen zu simulieren.

Die restlichen PCs dienen der Simulation der Windows-Umgebung, sollten jeweils einen Domaincontroller mit angeschlossenen Clients darstellen. Dabei war der Ausgangspunkt, die zentrale AD-Domain und eine angeschlossene Subdomain nachzustellen. Die Clients wurden jeweils mit WindowsXP-SP2, die Server mit Windows 2003 Server SP1 betrieben. Zu Debugzwecken stand weiterhin die Debug/Checked-Version von Windows 2003 Server und WindowsXP, sowie Windows 2003 Server ohne Service Pack zur Verfügung. Auf die Besonderheiten des Service Packs wird in der Folge nochmals eingegangen. Vom Einsatz von Windows 2003 in der Debug/Checked-Version habe ich mir auf Grundlage von [Ker04] mehr Einblick in die Ursachen von Kerberosfehlern erhofft, was sich in der Folge der Tests nicht bestätigte.

5.3 Intuitiver Versuch

Ein erster Versuch sah vor, jeglichen Netztraffic, der an das Windows-KDC gerichtet ist, an den Heimdal-KDC umzulenken und nur das eigentliche Verzeichnis auf dem Windows zu behalten. Dazu wurde im XP-Client ein Linux-Rechner als Domaincontroller angegeben und auf diesem mittels *iptables* und auch verschiedenen UDP-Proxys der Traffic auf den LDAP-Ports an den Windows-Server weitervermittelt. Dies hätte den Vorteil gehabt, die Konfiguration der Clienten einfachst zu halten und eigentlich serverseitig sämtliche Vorkehrungen zur Integration umzusetzen.

Zur Leitung des Traffics an den Linux-Rechner wurde ausgenutzt, dass Windows XP seine Ressourcen mittels DNS lokalisiert. Die entsprechenden SRV-Einträge wurden auf den Linux-Rechner gemapt, wodurch der Traffic gelenkt wurde. In der Folge wurde zusätzlich mittels der Windows Support Tools¹ versucht, diese Einstellungen clientenseitig zu manifestieren. Wie später auch, wurde hierfür das *ksetup*²-Tool eingesetzt.

Da einerseits eine grosse Zahl an Principals für die Dienste im Active Directory existieren, andererseits diese automatisch regelmäßig mit einem neuen Passwort versehen werden, ist eine Pflege dieser auf einem externen Kerberos sehr aufwändig, wenn nicht sogar unmöglich zu handhaben.

Ausgehend von diesem Ergebnis wurde strukturiert an die Konstruktion einer Lösung nach den Beschreibung von der University of Michigan in den USA unter [Mic05] oder der University of Alberta in Kanada unter [Alb05] herangegangen.

5.4 Namensraum - Vereinheitlichung der Namensgebung

Ein wesentlicher Aspekt, der den Betrieb von Windows Domänen innerhalb des Netzes des Universitätsrechenzentrums vereinfacht, ist eine einheitliche Regelung zur Namensgebung für Domänen. Diese Regelung sollte grundsätzlich gefunden werden, unabhängig davon ob eine Gesamtstruktur durch Bereitstellung einer Active Directory Domain als Dienst durch das URZ angestrebt wird oder ob weiterhin ein Betrieb von Insellösungen angedacht wird, da sich Vor- und Nachteile eines solchen Dienstes aufwiegen.

Grundlegend wollen die Domaincontroller einer AD-Domain Master innerhalb ihres Namensraumes sein, es können also innerhalb eines Namensraumes nicht mehrere Domänen betrieben werden. Auch bringt die Gliederung in klare Namensräume Klarheit in strukturelle Gedanken für den Betrieb bestimmter Dienste oder die Bereitstellung von Ressourcen durch Windows Domaincontroller.

Ursache dieser Konvention ist die Nutzung von DNS zur Lokalisierung von Diensten und Domaincontrollern durch die Windows-Betriebssysteme.

Auf Grund dieser Überlegungen ist es sinnvoll, von dem bisherigen Konzept der Strukturierung aller Rechner bestimmter Struktureinheiten (beispielsweise Fakultäten oder Institute) geringfügig abzuweichen und die vorhandene Struktur ein wenig zu erweitern.

Die Clienten einer Active Directory Domain können bei den Überlegungen außen vor gelassen werden, da deren DNS-Suffix unabhängig derer Domainzugehörigkeit ist. Die Do-

¹Jeweils auf der InstallationsCD der Betriebssysteme als zusätzliches Paket enthalten.

²Bestandteil der Support Tools, bei WindowsXP nur in der vollständigen Paket-Installation enthalten.

main, der der Rechner angeschlossen ist, wird in der Registry vermerkt und über diesen Vermerk die Lokalisierung der Domainsdienste reguliert. Somit kann ein Client den DNS-Suffix „sub1.domain.tld“ haben, aber der Domäne „sub2.domain.tld“ angehören. Einzig die standardmäßige Verknüpfung der Anpassung des DNS-Suffixes an die zugehörige Domain muss deaktiviert werden. Dies ist bei den Erweiterten Namenseigenschaften eines Clienten ein einzelnes Häkchen im System-Feld der Systemeigenschaften, welches standardmäßig gesetzt ist und entfernt werden muss.

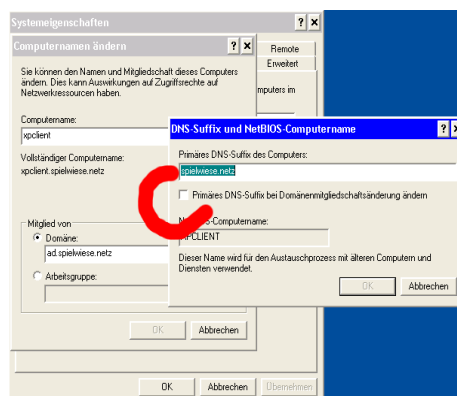


Abbildung 5.3: Trennung von DNS-Suffix und Domainnamen

Die betreffenden Systeme sind also die Domaincontroller einer Domäne, welche in ihrem eigenen Namensraum lokalisiert werden müssen. Auf einem Domaincontroller ist der Name der Domain fest mit dem DNS-Suffix verbunden. Wird ein Windows Server zum Domaincontroller deklariert, macht er als erstes einen Test, der den DNS-SRV-Record für den LDAP-Dienst der angegebenen Domain testet. Wird dabei ein zweiter Rechner neben seinem eigenen Namen geliefert und angegeben, dass er eine eigene oder andere als der gefundenen Domain kontrollieren soll, scheitert das Einrichten der Domain. Daher muss also für jede einzurichtende AD-Domäne ein eigener Namensraum geschaffen werden.

Ein weiterer zu beachtender Punkt ist der Umstand, dass Windows bei hierarchisch gegliederter Domainstruktur automatisch auf eine Subdomänenstruktur schließt und versucht beziehungsweise davon ausgeht, dass eine Vertrauensstellung besteht. Sollte also eine zentrale Baumstruktur angedacht werden, die etwa an der Wurzel „ad.tu-chemnitz.de“ lautet, wird bei einer Domäne „sub.ad.tu-chemnitz.de“ automatisch von einer Subdomäne ausgegangen. Dabei ist Subdomäne nicht nur im Sinne von DNS gemeint, sondern im Sinne von Active Directory.

Solange keine Notwendigkeit von äquivalenten Software Resource Records im zentralen DNS dagegen sprechen, kann eine AD-Domäne auch im selben Namensraum liegen wie andere Unix-Dienste oder Rechner ohne Domaincontrollerfunktionalität. So ist etwa eine zentrale Domäne mit dem Namen „tu-chemnitz.de“ denkbar, verursacht aber die stete Annahme aller anderen Domänen als Subdomänen. Somit könnte eine Domain names „ad.tu-chemnitz.de“ oder „windows.tu-chemnitz.de“ eine günstige Lösung darstellen, unterhalb derer eine Subdomänenstruktur auf Windows-Basis wachsen kann. Grundsätzlich bietet sich

zum Beispiel die Übernahme der Struktureinheitenstruktur als Namenskonvention an, so dass eine Fakultät eine Domain mit dem Namen „fakultaet.tu-chemnitz.de“ betreiben kann, ein Lehrstuhl der Fakultät „lehrstuhl.fakultaet.tu-chemnitz.de“. Auch beim Aufbau einer solchen Struktur ist mit den Randbedingungen zu arbeiten, die hier bereits erwähnt wurden und noch folgen.

Ein sinnvoller Ansatz der Namensraumgestaltung, der sowohl eine einheitliche Baumstruktur als auch Inselösungen unterstützt, wäre die Schaffung eines oben bereits angesprochenen Namensraumes für Windowsdomänen „ad.tu-chemnitz.de“, unter dem ein Subdomänenbaum mit „fakultaet.ad.tu-chemnitz.de“ und „lehrstuhl.fakultaet.ad.tu-chemnitz.de“ sowie den damit verbundenen Vertrauensstellungen, aber auch Einschränkungen, aufspannt. Inseln könnten dann beispielsweise zur Kennzeichnung als Windowsdomains in einem Raum namens „ad.lehrstuhl.fakultaet.tu-chemnitz.de“ stehen.

Der Namensraum gibt Randbedingungen für das DNS vor, so dass darauf in der Folge eingegangen werden soll.

5.5 DNS - dynamisch versus statisch

Eine grundlegende Frage, die sich stellte, aber auch recht zügig und abschließend beantwortet werden kann, ist die nach den Vorgaben an das DNS-System, welches für die Windows-Umgebung bereitgestellt werden muss.

Grundsätzlich wird darauf hingewiesen, dass das Konzept von Active Directory ein dynamisches DNS-System (DDNS) voraussetzt, um die Wartung und Pflege zu vereinfachen und die große Zahl an Einträgen zu handhaben. Dieses System muss vor allem Software Resource Records (kurz SRV-Einträge) unterstützen, um die Lokalisierung von Diensten auf Basis des DNS-Dienstes zu ermöglichen. Außerdem ist wohl eine Intension des DDNS, dass Clienten, die ihre Netzwerkeinstellungen per DHCP erhalten, ihre DNS-Einträge aktuell halten können. Wie im Theorieteil bereits angesprochen, werden standardmäßig eine Unzahl von Einträge erstellt, was wahrlich einen größeren Aufwand bei der Pflege eines statischen Systems bedeutete. Da innerhalb dieser aber viele Redundanzen an Informationen liegen, verringert sich der Bedarf an Einträgen auf eine überschaubare Anzahl notwendiger, wenn ein paar wenige DNS-Anfragen mehr kein Problem in der Netzkapazität darstellen beziehungsweise die Struktur der Domänen nicht durch viele Standorte mit zum Teil sehr schwacher Netzwerkanbindung geprägt ist.

Ein Grund für den Einsatz dynamischer Updates, der einem im Kontext von serverbasierten Diensten in verteilten Umgebungen einfällt, die Unterstützung von Failover-Szenarien, hat sich aus den Versuchen in diesem Zusammenhang nicht bestätigt. Die verschiedenen DNS-Einträge, die Dienste referieren, werden bei Ausfall eines Domaincontrollers nicht aktualisiert oder an die neue Umgebung angepasst. Vielmehr müssen Betriebsmasterausfälle manuell durch Verschieben der Rolle kompensiert, GC-Server-Ausfälle können durch redundante Bereitstellung abgefangen werden. Bei statischem DNS ist demnach nur ein Verschieben der PDC-Rolle nachzuvollziehen.

Aus der Kombination dieser Erkenntnis mit dem Umstand, dass im Netz des URZ eine feste Zuordnung von MAC- zu IP-Adressen stattfindet, somit das dynamische Update durch den Clienten hinfällig ist, lässt sich das Maß der notwendigen DNS-Einträge einschränken. Ein Einsatz von dynamischen Updates wird damit als nicht notwendig erachtet. Inwieweit sich

dadurch die in der Literatur angesprochene Performanceeinbuße bemerkbar macht, kann hier nicht eingeschätzt werden, da sowohl die Netzgegebenheiten als auch die Rechneranzahl in der Testumgebung eine echte Einschätzung nicht zulässt.

Natürlich ist davon auszugehen, dass die recht große Anzahl der Windowsclients im gesamten Campusnetz eine Lastverteilung auf alle im Netz verfügbaren Domaincontroller und insbesondere bei der Authentifizierung auf die GC-Server sinnvoll macht, eventuell auch die ortsgebundene Strukturierung über Campusteile. Die Werkzeuge des BIND-Systems oder anderer Netzbeeinflussungen bieten genügend Möglichkeiten dem gerecht zu werden. Der Einsatz von RoundRobin bei Anfragen oder Zuweisung eines GC-Servers je nach Subnetz wären hier denkbar.

Prinzipiell sind zwei Möglichkeiten der Umsetzung eines DNS-Konzepts denkbar, die beide im Rahmen der Teststellung getestet wurden. Die eine Möglichkeit besteht aus der Erstellung einer gesonderten DNS-Domäne auf dem externen DNS-Server (BIND), welche neben den Hosteinträgen auch die notwendigen SRV-Records enthält und somit bereitstellt. Diese Variante ist der Aufgabenstellung folgend die sinnvollste zum Betrieb einer zentralen AD-Domäne.

Beim Betrieb weiterer Subdomänen beispielsweise ist eine weitere Konstellation der DNS-Struktur denkbar. So wird für die Domäne im zentralen DNS-System nur ein Verweis auf einen domainspezifischen DNS-Server angelegt und innerhalb der Domäne wird ein eigenständiger DNS-Server betrieben, im Kontext der Aufgabe vorzugsweise ein MS-DNS-Dienst. Dadurch kann die Administration von Domänenspezifika an diese delegiert werden. Etwa ist denkbar, dass ein zusätzlich auf dem Domaincontroller betriebener Dienst weitere Einträge im DNS der Domäne hinterlegt. Derartige Einträge können dann durch den Administrator der Subdomäne gepflegt werden. Auch kann an dieser Stelle keine Aussage über die Notwendigkeit von weiteren SRV-Einträge getroffen werden, die nicht im Zusammenhang mit dem grundlegenden Domaincontroller-Dienst stehen.

Ausgehend von der im Theorieteil aufgeführten Tabelle können folgende Einträge für den Betrieb einer Domäne, wie es bei einer zentralen AD-Domäne möglich wäre, generell als notwendig erachtet werden:

<u>_msdcs.domain.tld.</u>					
_ldap._tcp.dc	SRV	0	100	389	<i>Domaincontroller</i>
_kerberos._tcp.dc	SRV	0	100	88	<i>Domaincontroller</i>
<hr/>					
_ldap._tcp.gc	SRV	0	100	3268	<i>Domaincontroller</i>
_ldap._tcp.pdc	SRV	0	100	389	<i>Domaincontroller</i>
<hr/>					
domain.tld.					
_ldap._tcp	SRV	0	100	389	<i>Domaincontroller</i>
_kerberos._tcp	SRV	0	100	88	<i>Domaincontroller</i>
_gc._tcp	SRV	0	100	3268	<i>Domaincontroller</i>
_kpasswd._tcp	SRV	0	100	464	<i>Domaincontroller</i>
<hr/>					
_kerberos._udp	SRV	0	100	88	<i>Domaincontroller</i>
_kpasswd._udp	SRV	0	100	464	<i>Domaincontroller</i>
<hr/>					
gc._msdcs	A				<i>IP des Domaincontrollers</i>
<i>Domaincontroller</i>	A				<i>IP des Domaincontrollers</i>

Im Anhang A findet sich die beispielhafte Konfigurationsdatei des BINDs aus dem Betrieb einer AD-Domäne im Rahmen der Teststellung. Dabei ist „ad.spielwiese.netz“ eine mittels statischem DNS konfigurierte Domäne, „sub.ad.spielwiese.netz“ eine Subdomäne, die einen eigenen DNS-Server besitzt, auf welchen nur mittels eigener Zone und dem NS-Eintrag in der ad.spielwiese.netz-Zone verwiesen wird. Die SRV-Einträge in der Zone „spielwiese.netz“ dienen der Abbildung von Hostnamen auf Kerberos-Realms und deren Dienstlokation. Auch Heimdal und MIT unterstützen die Lokalisierung ihrer Ressourcen per DNS. Wird sich im Rahmen der Strukturdebatte für die komplette Zentralisierung der DNS-Verwaltung ausgesprochen, müssen auch die DNS-Einträge der Subdomänen eingepflegt werden. Ausgehend von der im Theorieteil aufgezeigten Tabelle sind folgende Einträge als notwendig anzusehen:

sub.domain.tld.					
_ldap._tcp.dc._msdcs	SRV	0	100	389	Subdomaincontroller
_kerberos._tcp.dc._msdcs	SRV	0	100	88	Subdomaincontroller
_ldap._tcp.pdc._msdcs	SRV	0	100	389	Subdomaincontroller
_ldap._tcp	SRV	0	100	389	Subdomaincontroller
_kerberos._tcp	SRV	0	100	88	Subdomaincontroller
_kpasswd._tcp	SRV	0	100	464	Subdomaincontroller
_kerberos._udp	SRV	0	100	88	Subdomaincontroller
_kpasswd._udp	SRV	0	100	464	Subdomaincontroller
Subdomaincontroller	A				IP des Subdomaincontrollers

Der Eintrag der DomainID, der im „_msdcs“-Bereich der Root-Domäne eingetragen wird, entfällt, da keinerlei ID-Einträge übernommen werden.

Alternativ, wie bereits angesprochen, ist es ebenfalls möglich, seitens der zentralen Struktur nur einen Verweis auf den zuständigen DNS-Server für die dezentrale Domäne einzurichten. Somit bliebe es in der Verantwortlichkeit der Subdomänenadministration, ob dort dynamisch oder statisch die DNS-Einträge gepflegt werden und der Betreiber eines solchen Windows-DCs in einer Struktureinheit kann entscheiden, ob er beispielsweise einen MS-DNS-Server auf seinem Domaincontroller betreibt.

Dazu wird zentral für die Domäne eine eigenständige Zone eingerichtet, die vom Typ „forward“ ist und durch „forward first“ die Anfragen direkt an den definierten Server weiterleiten. Ist die Domäne eine Substruktur einer auf dem zentralen System als „master“ definierten Zone, ist in dieser zusätzlich ein „NS“-Eintrag für die Subdomäne einzutragen.

Dies könnte wie folgt aussehen:

```
zone "sub.domain.tld" IN {
    type forward;
    forward first;
    forwarders { DNS-Server-IP; };
};
```

Es bleibt also festzuhalten, dass der grundlegende Betrieb einer AD-Domäne auch mit einem statischen externen DNS-System realisierbar ist. Auf dem externen System werden eine Unzahl von Log-Einträgen erzeugt, die auf den abgewiesenen Versuch hinweisen, durch einen Windows-Rechner seinen DNS-Eintrag dynamisch zu aktualisieren.

Inwieweit eine komplette Zentralisierung der DNS-Technologie für die Windows-Systeme gewünscht oder vorangetrieben wird, ist eine politische Entscheidung, die im Zusammenhang mit der zu führenden Strukturentscheidung zu sehen ist.

5.6 Kerberos im genannten Kontext

Vieles ist bereits im Theorieteil (siehe 2.2.1) zu Kerberos genannt worden und soll hier als Grundverständnis vorausgesetzt werden. Aufbauend finden sich hier einige Eigenarten und Bemerkungen zur Implementierung der Windows 2003 Server und zum Zusammenspiel einer Windows-Umgebung mit einem externen Kerberos-Realm.

Ein Punkt, welcher in der Literatur bei [Gar03] genannt wird, in meinen Versuchen aber keinerlei Einfluss deutlich zeigte, ist die Erweiterung der AS-Response um das so genannte PAC, das Privilege Access Certificate. Das PAC wird als Anhang des TGTs ausgeliefert und enthält die SID und Gruppenrechte des authentifizierenden Nutzers. Dadurch wird versucht, die Anfragenmenge in Folge einer Anmeldung zu reduzieren. Wird das PAC nicht ausgeliefert, werden die enthaltenen Informationen durch den Clienten aus dem LDAP erfragt. Es entsteht also ein Overhead von einer oder wenigen LDAP-Anfragen durch einen Nicht-Windows-KDC. Probleme entstanden durch diesen Sachverhalt allerdings mit älteren Nicht-Windows-Kerberos-Clienten in der Vergangenheit aus dem Punkt, dass die Gruppenrechte natürlich recht umfangreich sein können, wodurch ein UDP-Paket, welches üblicherweise das TGT liefert, nicht ausreicht, um das TGT und PAC zu liefern. Daher fordert in solchen Fällen ein Windows-KDC den Clienten auf, einen TCP-Retry durchzuführen. Ältere Kerberos-Clienten können dies nicht, da der Standard ursprünglich nur UDP als Protokoll vorsah. Da die Klienten in der hier gemachten Betrachtung hauptsächlich Windows-Systeme sind und die neueren Implementierungen der Heimdal- und MIT-Entwicklungen ebenso TCP-Anfragen unterstützen, soll hier nicht weiter darauf eingegangen werden.

Generell ist zu erwähnen, dass Literatur, die die Integration von Windows und anderen Systemen anpreist, in den überwiegenden Fällen auf eine Zentralisierung mit Windows Servern im Kern hinausläuft und die Datenübernahme aus anderen üblichen Quellen in ein Active Directory und den Anschluss verschiedener Clientensysteme an das Windows System beschreibt. Typische Vertreter, deren Titel mich anderes vermuten ließen, sind [Sch03] und [Bod05]. Dass dies offenbar nach Vorstellung vieler Seiten der sinnvollere Weg ist, beschreibt ebenfalls das MS Technet Paper [Win00b].

So lassen sich beispielsweise auch die Keytabs, die ein Unix-Kerberos-System als Client für das Service- oder Host-Principal nutzt, aus dem Windows KDC per *kt pass*-Werkzeug generieren und Schlüssel auf diesem Weg exportieren. Hingegen ist eine Importfunktion, obgleich in der Hilfe erwähnt, nicht implementiert. Somit lassen sich nicht einfach vorhandene Schlüssel aus einem externen Kerberos-System in einen Domaincontroller importieren.

5.6.1 Windows und Schlüsseltypen

Hinsichtlich der Verschlüsselungen, die der Windows Domaincontroller akzeptiert, wurde bereits erwähnt, dass nur RC4- und DES-Schlüssel mit MD5- und CRC-Hash verarbeitet

werden können. Heimdals RC4³-Implementierung ist allerdings nicht mit Windows kompatibel, der Domaincontroller ist nicht befähigt, ein derart verschlüsseltes Service Ticket zu validieren. Somit kommen für die Kooperation von Heimdal und Windows nur DES-Schlüssel in Frage. Dabei ist darauf zu achten, dass nicht nur die Kommunikation in DES verschlüsselt wird, was unkompliziert ist, da DES die Standardverschlüsselung von Heimdal ist und von Windows-Clients auch angefordert werden. Auch die Tickets müssen mit DES-Verschlüsselung ausgeliefert werden, was durch die Löschung der entsprechenden Typen aus der Principal-Datenbank oder durch Setzen des bevorzugten Standardschlüsseltypen in der *krb5.conf* erreicht werden kann. Erkennbar wird dieses Problem durch einen entsprechenden Fehler in der Kerberos-Kommunikation. Schwieriger ist der Fehler zu erkennen, wenn die Kommunikation zur Erlangung des Tickets korrekt verläuft, das erworbene Ticket allerdings nicht auswertbar ist und der Client dann mit einem PRINCIPAL-UNKNOWN-Error reagiert. Dies ist im Laufe der Tests mit einem MIT-KDC geschehen, der standardmäßig eine 3DES-Verschlüsselung verwendet und erst durch explizites Deaktivieren und Löschung der entsprechenden Principal-Einträge zu einer DES-Kommunikation gebracht werden konnte. Die Kommunikation zur Erlangung des TGTs funktionierte und das TGT war auch akzeptabel, allerdings das für den Cross-Realm-Trust ausgegebene TGT für die Windows-Domain wurde mit AES verschlüsselt, was dann am Domaincontroller scheiterte.

5.6.2 Windows mit externem Kerberos

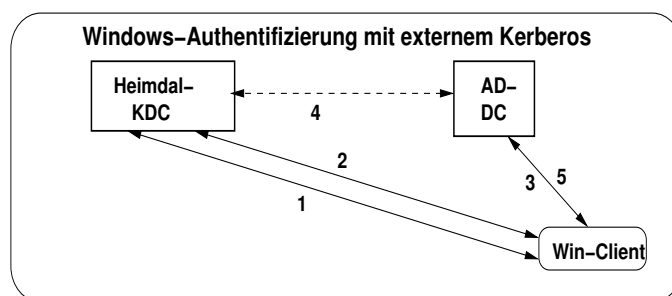
Eine funktionsfähige Zusammenarbeit eines Windows-Systems mit einem externen Kerberos funktioniert nur, indem mit dem *ksetup*-Kommandozeilenwerkzeug aus den Support Tools, welche auf der Windows-Installations-CD als Zusatzpaket mitgeliefert werden, ein KDC für einen externen Realm angegeben wird. Somit wird, wenn im Anmeldebildschirm unter „Anmelden an“ der externe Realm ausgewählt ist, der Nutzer gegen den externen KDC verifiziert und durch diesen das TGT erteilt. Ist der Rechner, an dem der Nutzer sich anmeldet, Mitglied einer Domain unterscheidet sich der Weg ab diesem Punkt.

Der Client versucht ein Service-Ticket für seinen Rechner zu erlangen und wird an diesem Punkt vom externen KDC an seinen Domaincontroller per Realm-Trust verwiesen. Dieser erteilt das Ticket. Nun werden ein LDAP-Ticket angefordert und mit diesem ein Nutzer in der Domain gesucht, der namensgleich dem Nutzerprincipal ist und ein Attribut hat, welches ihn auf das Kerberos-Principal abbildet. Dieses Attribut heißt „altSecurityIdentities“ und kann sowohl alternative Windows-Accounts als auch Kerberos-Principals aufnehmen. Wird der Nutzer in der Domain gefunden, wird dieser am Clienten angemeldet und alle Berechtigungen und Einschränkungen des Domainaccounts wirksam. Den Ablauf einer solchen Anmeldung eines Nutzers an einem Windows-Clients, der gegen einen externen Kerberos-Server authentifiziert wird, verdeutlicht Abbildung 5.4 nochmals.

Ist der Rechner hingegen nicht in einer Domäne, wird dieses Mapping gegen die vorhandenen lokalen Accounts durchgeführt und versucht, ein Nutzer zu finden, der exakt so heißt wie das Nutzer-Principal oder ein passendes User Mapping gesucht. User Mapping heißt, dass eine Abbildung fremder Nutzernamen auf lokale Accounts stattfindet. Dies kann ebenfalls mittels *ksetup* eingestellt werden.

An dieser Stelle wird auch ein Aspekt wirksam, der oben bereits angedeutet wurde und im

³dort ARC4 oder arcfour genannt



1. Nutzer meldet sich an "TU-CHEMNITZ.DE" mit NKZ & Passwort an, Windows-Client erlangt TGT fuer "TU-CHEMNITZ.DE"-Realm
 2. Windows-Client verlangt, TGS fuer seinen Host Heimdal kann dieses nicht erteilen, verweist aber an AD-Server
 3. Windows-Client verlangt TGS fuer seinen Host unter Vorlage des "TU-CHEMNITZ.DE"-TGT
 4. AD-Server validiert TGT via Cross-Realm-Trust (Vertrauensstellung)
 5. AD-Server erteilt TGS fuer Windows-Clients
- Auf selbem Weg (3.-5.) sucht Windows-Client nach Nutzer im LDAP, der zum Prinzipal passt. Nutzeraccount mit passender Namenszuordnung wird an der Domain angemeldet.

Abbildung 5.4: Anmeldung eines Kerberos-Nutzers an einem Windows-Clienten

Zusammenhang mit dem auf [Win05] vorgestellten Windows 2003 Server Service Pack steht. Gerade dieses Service Pack des Servers brachte Neuerungen mit, die die Zusammenarbeit mit einem externen Kerberos erschwerten – der vormals mögliche Betrieb mit deaktivierten Accounts und Kerberos wurde im Service Pack mittels Registry-Eintrag unterbunden, so dass nun das Deaktivieren der Accounts sich auch auf Nutzer auswirkte, welche per Kerberos authentifiziert werden. Vormals war dies anders, wodurch ein Missbrauch durch die Nutzung des Windows-Passwortes des Accounts verhindert werden konnte. Durch die Aktivierung des Accounts gibt es natürlich auch die Möglichkeit, sich an der Domäne mit dem Passwort zu authentifizieren, welches im Domaincontroller gespeichert ist. Somit ist auch darauf zu achten, ein sicheres Passwort für den Windows-Account zu setzen, um Missbrauch auf diesem Wege zu verhindern.

Was also notwendig ist, um eine Windows-Domäne nutzerseitig zur Authentifizierung an einen externen Kerberos-Dienst zu binden, ist die Installation der Support Tools auf allen Clienten der Domäne und deren Einrichtung eines zusätzlichen Realms dessen KDC der externe Kerberos-Server ist. Es können dort auch mehrere Server angegeben werden. Dies geschieht normalerweise mittels des Support-Tools „ksetup“, welches nur bei der vollständigen Installation der selbigen enthalten ist. Die Syntax ist einfach, kann mittels eines Aufrufs mit der Option „/?“ ermittelt werden, findet sich aber auch unter anderem auf [kse03] zum Nachlesen. Dort finden sich auch Beispiele der Nutzung. Andererseits ist die Installation des Werkzeugs auf allen betreffenden Rechnern recht aufwendig, nur um eine

einmalige Einrichtung vorzunehmen. Ksetup erstellt nur einige wenige Registry-Keys, die auch mit selber Wirkung in die Registry eingetragen werden können. Es wird im Baum `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos_Domains` ein Schlüssel angelegt, der dem Namen des Kerberos-Realms entspricht, also beispielsweise „TU-CHEMNITZ.DE“. In diesem wird ein Wert namens „KdcNames“ vom Typ „REG_MULTI_SZ“ erstellt, der die DNS-Namen der KDC-Server für den entsprechenden Realm enthält. Unangenehm ist der Umstand, dass Werte dieses Types beim Export hexadezimal abgelegt werden. Sind für den Realm weitere Flags mittels des Werkzeugs eingetragen, werden diese als Wert des Typs „REG_DWORD“ unter der Bezeichnung „RealmFlags“ abgelegt und sind auch beim Export lesbar.

Außerdem lässt sich mittels „ksetup“ ein Mapping von externen sich anmeldenden Nutzern auf lokale Accounts einrichten. Dazu wird in der Registry unter `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos` ein Schlüssel namens „UserList“ erstellt, der dann das zu mappende externe Prinzipal als Wert vom Typ „REG_SZ“ enthält, welcher wiederum den Zielaccount fasst. Somit ließe sich vermutlich der Aufwand der Installation der Support Tools und der Ausführung von *ksetup* durch ein exportiertes Registry-File ersetzen. Andererseits werden die Support Tools als msi-Paket bereitgestellt, was eine Installation im Rahmen der Grundinstallation oder auch mittels Gruppenrichtlinie später vereinfacht.

Um nun die externe Kerberos-Authentifizierung mit einer Domäne zu nutzen, muss zusätzlich zwischen dem Kerberos-Realm und der Windowsdomäne eine Vertrauensstellung, bei Kerberos „Cross-Realm-Trust“ genannt, eingerichtet werden. Da hier nur vorgesehen ist, dass sich Kerberos-Nutzer an der Windowsdomäne anmelden können, der Umkehrschluss nicht erwünscht ist, reicht eine ausgehende Vertrauensstellung vom Windows aus gesehen. Dazu muss im externen Kerberos ein Principal angelegt werden, welches die Bezeichnung *krbtgt/windows.domain.tld* hat und kein zufälliges Passwort besitzt. Auf einem Domaincontroller der Domäne wird im Gegenzug in den „Active Directory Standorten und Vertrauensstellungen“ eine solche ausgehende Vertrauensstellung zum externen Kerberos-Realm eingerichtet, welche das dem TGT zugewiesene Passwort nutzt. Damit ist die Zusammenarbeit grundlegend eingerichtet. „Nur“ die Pflege der Accounts bleibt, die mit den Kerberos-Principals korrespondieren.

Nachdem der Nutzer gegen den externen Kerberos-Server authentifiziert wurde, versucht der Client auf das Active Directory zuzugreifen, um die Daten des Nutzers zu erfahren und Einstellungen für den Desktop des Nutzers zu übernehmen. Dazu erfragt er beim externen Kerberos ein Ticket für den Dienst „LDAP/Domaincontroller“, was mittels des Realm-Trusts zu einem Verweis an den Domaincontroller führt, da der KDC dieses Ticket nicht erteilen kann. Hier wird auch ein Problem ersichtlich, alles auf einem externen KDC pflegen zu wollen, da der Schlüssel, der auf dem Domaincontroller dem Dienst zugeteilt wird, nicht bekannt ist, also kein gültiges Service-Ticket erteilt werden könnte. Auch aktualisiert der Domaincontroller die Schlüssel seiner Dienste im Active Directory regelmäßig.

5.6.3 Heimdal– versus MIT-Kerberos

Neben der bereits mehrfach erwähnten schwedischen Kerberos-Implementierung genannt „Heimdal“ gibt es eine zweite am amerikanischen Massachusetts Institute of Technology

beheimatete Implementierung, welche auf [MIT05] gefunden werden kann.

Die beiden frei zugänglichen Kerberos-Implementierungen, die als externes Kerberos-System zum Einsatz kommen könnten, unterscheiden sich einerseits in der Liste und Implementierung der unterstützten Schlüsseltypen, aber auch in der Behandlung der Anfragen.

Auf den erstgenannten Umstand soll hier nur verwiesen werden, da keine umfangreichen Tests mit der MIT-Implementierung stattgefunden haben, insbesondere die ARC4-Verschlüsselung wurde nicht auf Kompatibilität mit der Microsoft-Implementierung hin untersucht. Allerdings machte die Unterscheidung bei der Behandlung der Anfragen einen Test beider Implementierungen interessant.

Beim Betrieb einer Subdomäne stellte sich im Rahmen der Versuche heraus, dass entgegen den Beschreibungen aus den Quellen [Mic05] der University of Michigan und [Alb05] der kanadischen University of Alberta nicht einem Trust-Path über die Root-Domain gefolgt wird. Bestand ein Realm-Trust zwischen Heimdal und der Subdomäne wurde direkt nach dem Authentifizierungsschritt durch den externen Heimdal-Kerberos an den Domaincontroller der Subdomäne verwiesen. Dieser kann auch das Service-Ticket für den Client-Rechner bereitstellen. Allerdings wird im Folgeschritt versucht, einen Domainbenutzer zu finden, der dem Prinzipal des externen Kerberos entspricht. Ist dieser vorhanden, kann der Nutzer mit den Berechtigungen dieses Domainbenutzerobjektes angemeldet werden. Ziel der Aufgabenstellung ist allerdings eine zentrale Nutzerverwaltung, in der die Nutzer an nur einer Stelle verwaltet werden sollen. Auch ist eine Einschränkung des Zugriffs auf personenbezogene Daten auf einen überschaubaren Personenkreis zu beachten.

Entsprechend der oben genannten Quellen sei eine Bereitstellung der Nutzer im Active Directory der Root-Domäne ausreichend, da der Prinzipal mit dem ersten Vorkommen eines passenden Nutzerobjektes auf dem Trust-Path angemeldet wird. Dies entspricht allerdings nicht dem beschriebenen Verhalten des Heimdal-Kerberos. In den Quellen wurde erwähnt, dass einzig eine Vertrauensstellung zwischen der Windows-Root-Domäne und dem externen Kerberos bestehen muss. Ähnliches wird in verschiedenen Papers der University of Colorado unter [Col00] beschrieben. Ist dies bei der Heimdal-Implementierung der Fall, wird die Anfrage des Klienten nach seinem Service-Ticket durch den Heimdal-Server einzig durch einen „KRB5_ERROR_PRINZIPAL_UNKNOWN“ beantwortet, da ihm auch kein Trust bekannt ist, der die Anfrage befriedigender bearbeiten kann.

Ein weiterer Versuch bestand darin, dieses Verhalten durch ein geschicktes Verschachteln von Gruppen eine Art Verweis auf die Nutzerobjekte in der Root-Domain innerhalb der Subdomäne anzulegen. Prinzipiell kennt X.500 ja Verweise, um auf den Datenbestand anderer DSA zu vermitteln, welche vom angesprochenen DSA nicht beantwortet werden können. Leider ist das Anlegen eines solchen Verweises in Form eines Nutzerobjektes innerhalb des Active Directorys nicht vorgesehen, weshalb auf die Variante der Gruppenmitgliedschaften ausgewichen werden sollte. Vertiefend könnte ein Versuch in dieser Richtung mit Hilfe von Visual Basic in der Folge der Arbeit durchgeführt werden, da allerdings die Erfolgsaussichten gering sind, wurde es an dieser Stelle nicht ausgeweitet. Eine sinnvolle Gruppenverschachtelung ist im Rahmen der Tests ebenfalls erfolglos geblieben, aber es sei auf Abschnitt 5.8 verwiesen, welcher verschiedene Erkenntnisse zu Gruppenarten und -verweisen enthält, die sicher auch für die tägliche Arbeit interessant sind.

Wie sich im Rahmen der Tests herausstellte, unterstützt hingegen MIT-Kerberos mittels einer Funktion genau eine Befolgung des oben genannten Szenarios. Ein AS-Request, wie er

bei der Erstauthentifizierung eines Nutzers gestellt wird, wird auch durch das MIT-Kerberos nur durch Abgleich des angeforderten Prinzipals mit der lokalen Prinzipaldatenbank abgearbeitet. Hingegen wird in der Funktion zur Abarbeitung von TGS-Requests „do_tgs()“ bei einem erfolglosen Versuch, das Prinzipal in der lokalen Datenbank zu finden oder einem direkten Trust folgend weiter zu vermitteln, eine Funktion gerufen, die das angeforderte Prinzipal an Trennzeichen (üblicherweise dem Punkt) zerlegt und dann die verschiedenen Kombinationen gegen die lokale Datenbasis abgleicht. Die Abweisung eines solchen Verhaltens bei AS-Requests erklärt sich durch das Kerberos-Protokoll, da die Weitervermittlung mittels Erteilung eines TGTs für den Zielrealm erfolgt, dessen Validierung aber durch den empfangenden Realm beim auslösenden KDC möglich sein muss. Beim AS-Request kann aber der ursprünglich angesprochene KDC noch niemanden validieren, da ihm ja eine Prüfung des Prinzipals nicht möglich war.

Beispielhaft am oben aufgeführten Szenario der Subdomain könnte das Verhalten des MIT bei einem TGS-Request wie folgt aussehen:

1. Ein Nutzer meldet sich von einem Windows-PC der Subdomain am MIT-Realm an. Der MIT-Kerberos erteilt das TGT für den Realm „krbtgt/MIT-REALM@MIT-REALM“.
2. Mit dem TGT versucht nun der Clienten-PC ein Ticket für den Dienst „host/clientpc.sub.domain.tld@MIT-REALM“ zu erhalten.
3. Da der MIT-Kerberos das Prinzipal nicht kennt und auch kein Cross-Realm-Trust zu „sub.domain.tld“ besteht, kann diese Anfrage nicht direkt bearbeitet werden.
4. Intern wird im MIT-Kerberos nun die Funktion *find_alternate_tgs* aus *do_tgs_req* aufgerufen, welche den Service-Teil des Requests abteilt und den Host-Bestandteil an den Punkten zerlegt und alle Kombinationen zurückliefert. Es wird also ein Feld erstellt, welches „host/sub.domain.tld@MIT-REALM“, „host/domain.tld@MIT-REALM“ und „host/tld@MIT-REALM“ enthält.
5. Auf Grund des Cross-Realm-Trusts, findet beim folgenden Abgleich der Feldinhalte mit der lokalen Datenbasis ein Verweis an die Windows-Root-Domain statt, welcher durch den Versuch „host/domain.tld@MIT-REALM“ einzufordern, ausgelöst wird. Dieser Verweis wird realisiert, in dem statt dessen ein „krbtgt/domain.tld@MIT-REALM“ ausgeliefert wird.

Auf diesem Wege wird auch die Auffindung eines zentral verwalteten Benutzerobjektes in der Windows-Root-Domain sicher gestellt. Einen ähnlichen Weg beschreibt ein Paper der University of Washington auf [Was01], welches die Verlagerung des Springens über eventuelle Realm-Referrals auf den Clienten beschreibt.

Da sich eben zeigte, dass der Aufbau eines Domänenbaums mit ausschließlichem Trust zur Wurzel domäne und dort zentral verwalteten Nutzeraccounts nur mittels des Einsatzes von MIT-Kerberos realisieren ließe, hingegen die derzeitige Dienststruktur des URZs nicht wesentlich geändert werden soll, wurde im Rahmen der Tests versucht, einen Heimdal-Master mit einem MIT-Slave zu koppeln und die Replikationsmechanismen der beiden Systeme zu kombinieren. Da offenbar allerdings die Propagation-Protokolle zur Replikation

von Änderungen in der Prinzipal-Datenbank des KDC-Masters auf die Slaves unterschiedlich implementiert, weil auch nicht standardisiert, wurden und Unterschiede in der Key-Implementierung hinzu kommen, war dieser Anschluss eines MIT- an ein Heimdal-System nicht erfolgreich. Somit ist es nicht möglich, innerhalb eines Baums mit externem Kerberos nur einmalig zentral Nutzerobjekte zu verwalten.

5.6.4 Passwortverwaltung mit externem Kerberos

Ein interessanter Aspekt, der sich im Kontext der Versuche herausstellte, war die Möglichkeit den Windows-Passwort-Änderungsmechanismus auch auf das externe Kerberos-System wirken zu lassen. Durch die Anmeldung des Nutzers am externen Kerberos-Realm und der damit verbundenen Vermittlung aller Kerberos-Anfragen primär an den KDC des externen Kerberos, wird auch eine Änderung des Passwortes durch den Nutzer an diesen KDC signalisiert, der die Änderungen entgegen nimmt und verarbeitet. Sowohl im Falle von Heimdal als auch von MIT-Kerberos führte ein solcher Versuch zu einem erfolgreichen Ändern des Prinzipalschlüssels in der Prinzipal-Datenbank, wodurch in der Folge auch das heimdal- beziehungsweise MIT-basierte *kinit* nur noch mit dem neuen Passwort erfolgreich verarbeitet wurde.

Vorteilhafterweise werden die Qualitätskontrollen des Windows-Systems, welche in einer Sicherheitsrichtlinie festgelegt werden, bei diesem Änderungsvorgang wirksam, so dass das Passwort bei Änderung über das Windows in der Standardkonfiguration komplexer sein muss, als beim Heimdal selbst.

5.6.5 Anmerkung zu OpenAFS & Windows Server 2003

Ungünstigerweise war es im Rahmen der Tests nicht möglich, auf den Windows Server 2003 Systemen ein gültiges Token für den OpenAFS-Clients in der Version 1.4.0 und auch 1.3.8 zu erhalten. Durch die Analyse des Netzwerkverkehrs war festzustellen, dass die Clients ausschließlich versuchten, Kerberos Tickets der Version 4 zu erlangen, was aus Sicherheitsgründen abgelehnt wird. Dieses Phänomen wurde auf allen in der Teststellung installierten Windows Server 2003 Instanzen beobachtet. Normalerweise sollte nach dem Scheitern des Versuchs Kerberos V4-Tickets zu erlangen, auf Kerberos V5 umgeschaltet und somit der Zugriff gewährt werden. Auf den Windows XP Instanzen der Teststellung hat dies auch unproblematisch funktioniert, allerdings war der Umstand auf den Server Systemen nicht nachvollziehbar. Dadurch war es auch mit Hilfe von Werkzeugen, wie dem „MIT Kerberos for Windows“, nicht möglich, ein gültiges Ticket zu erlangen und somit auf geschützte Bereiche des AFS zuzugreifen.

Da dieses Phänomen bisher nicht beobachtet wurde, wird die Möglichkeit in Betracht gezogen, dass die Probleme ursächlich im Zusammenhang mit der Firewallkonstruktion der Teststellung beziehungsweise eventuell einem bestimmten Update oder einer bei der Installation gemachten Einstellung stehen.

Wie ein nun schon mehrfach genanntes Nutzerobjekt aussehen könnte beziehungsweise welche Attribute für ein solches Objekt interessant wären, soll im folgenden Abschnitt be-

sprochen werden.

5.7 Benutzerverwaltung

Ein Benutzerobjekt in einer Domäne kann natürlich über das GUI eines Domaincontrollers unter dem MMC „Active Directory Benutzer und Computer“ angelegt werden. Da im Rahmen dieser Arbeit einerseits aber hunderte beziehungsweise mehrere Tausend solcher Nutzerobjekte gepflegt werden sollen, die andererseits wieder nicht mit Werkzeugen des Active Directory sondern in der MoUSE verwaltet und nur in die Sphären des Active Directory übernommen werden sollen, muss diese Übernahme natürlich automatisiert und sicher erfolgen, um den Betreuungsaufwand zu minimieren.

Neben dem genannten GUI gibt es verschiedene Wege auf das AD zuzugreifen, die bereits im Theorieteil angesprochen wurden. Dabei ist zu beachten, dass die LDAP-Schnittstelle nicht mächtig genug ist, da sicherheitsrelevante Attribute, wie etwa das Passwort, in der Verwaltung des SAMs (Security Account Manager) stehen und der Zugriff auf solche Elemente verweigert wird.

Somit blieb nur die Betrachtung mittels der „ds“-Werkzeuge auf der Kommandozeile oder mittels VB-Scripte. Die „ds“-Werkzeuge, also *dsadd*, *dsmmod*, *dsquery* und *dsdel*, erlauben die Manipulation von Objekten des Directory Services (daher sicher auch das „ds“ in den Namen). Aber in der Implementierung der Werkzeuge ist nicht nur die Auswahl der Objekte, sondern dann auch wiederum deren Auswahl an Attributen festgelegt, die bearbeitet werden können. Da ein Attribut des Benutzerobjektes benötigt wird, welches nicht in der Standardauswahl der „ds“-Werkzeuge enthalten ist, nämlich die Zuordnung eines Kerberos-Prinzipsals, ist auch deren Einsatz zum Datenabgleich mit den Daten der MoUSE nicht möglich.

Dem entsprechend bleibt nur der Einsatz eines VB-Scripts, welches den Zugriff auf alle notwendigen Benutzerattribute gewährt und die Daten aus einer definierten Quelle einliest.

Die Liste der Attribute eines Benutzerobjektes kann dem Schema der jeweiligen Gesamtstruktur entnommen werden. Dort muss auch eine entsprechende Anpassung stattfinden, sollten weitere Attribute, neue Objekte oder strukturelle Anpassungen des Schemas gewünscht werden. Eine Auflistung der Standardobjekte findet sich unter anderem aber auch zusammengetragen von Sakari Kouti und Mika Seitsonen für ihr Buch „Inside Active Directory“ auf [Act01]. An dieser Stelle sei nur eine Liste der notwendigen, für sinnvoll aus der Quelle MoUSE ermittelbaren und derzeit nicht sinnvoll zentral bereitstellbaren Attribute genannt:

- notwendig:
 - distinguishedName (dn) — üblicherweise liegen die Nutzerobjekte in $CN = Users, DC = domain, DC = tld$, können aber zur besseren Strukturierung auch in jede OrganisationalUnit verlegt werden, etwa $OU = Studenten, DC = domain, DC = tld$. Das Objekt muss nach dem Kerberos-Prinzipal benannt werden und wird als CN abgelegt.

- displayName — Der Anzeigename im Windowssystem (bspw. im Startmenü), zusammengesetzt aus Vor- und Nachname.
 - description — Eine Beschreibung des Objektes und da im GUI dies neben dem Objektamen steht, welcher hier nur das Nutzerkennzeichen hält, können durch das Setzen auf Vor- und Nachname einfacher Realpersonennamen gefunden werden.
 - givenName — Ist der Vorname.
 - sn — Ist der Nachname.
 - cn — Der Objektname, also wie beim DN erwähnt, das Nutzerkennzeichen, da identisch mit dem Kerberos-Prinzipal.
 - name — identisch zum CN, das Nutzerkennzeichen.
 - sAMAccountName — Der Name des Sicherheitsprinzipals im SAM, also auch das Nutzerkennzeichen.
 - userPrincipalName — Bestimmt den Windowsnutzer, wird also auf *Nutzerkennzeichen@Windowsdomain* gesetzt.
 - altSecurityIdentities — Beschreibt die Abbildung des Nutzerobjektes auf alternative Sicherheitsprinzipale. Hier wird es zur Abbildung auf den Kerberos-Prinzipal genutzt und auf „Kerberos:*principal@externenRealm*“ gesetzt.
 - memberOf — kommaseparierte Liste von Windows-Gruppen-DNs, wodurch die Mitgliedschaft in Sicherheitsgruppen geregelt wird.
- weiterhin sinnvoll:
 - profilePath — Pfad zum Windows-Profil
 - scriptPath — Pfad zum Login-Skript
 - homeDirectory — Freigabepfad zum Homeverzeichnis (auch als lokaler Pfad möglich, wodurch das folgende Attribut entfallen kann)
 - homeDrive — Laufwerksbuchstabe, der mit dem HomeDirectory-Pfad verknüpft wird.
 - title — Personentitel, soweit vorhanden
 - department — strukturelle Zuordnung der Person in der Universität
 - mail — E-Mailadresse des Nutzers
 - telephoneNumber — dienstliche Telefonnummer
 - ipPhone — dienstliche VOIP-Telefonnummer
 - facsimileTelephoneNumber — dienstliche Telefaxnummer
 - weniger interessant
 - derzeit nicht in MoUSE aktuell
 - * physicalDeliveryOfficeName — Raumnummer von bspw. Mitarbeiterräumen
 - eventuell auch statisch für alle Nutzer eintragbar:

- * `streetAddress` — Straße der Postanschrift
 - * `postOfficeBox` — Postfachnummer
 - * `postalCode` — Postleitzahl
 - * `l` — Ortsname
 - * `c` — Ländercode (bspw. DE)
 - * `co` — Landesname (bspw. Deutschland)
 - * `st` — Bundesland (bspw. Sachsen)
 - * `company` — Firmen- oder Einrichtungsname (bspw. Universität Chemnitz)
- derzeit weniger sinnvoll:
 - `wWWHomePage` — Webseite der Person
 - `pager` — Pagernummer
 - `homePhone` — private Telefonnummer
 - `mobile` — Mobiltelefonnummer
 - `url` — weitere Webseite
 - `otherTelephone` — weitere dienstliche Telefonnummer
 - `otherIpPhone` — weitere VOIP-Telefonnummer
 - `otherFacsimileTelephoneNumber` — weitere Telefaxnummer
 - `otherHomephone` — weitere private Telefonnummer
 - `otherMobile` — weitere Mobiltelefonnummer
 - `otherPager` — weitere Pagernummer
 - *Sämtliche kursiv geschriebenen Attribute können im Objekt beliebig oft vorkommen.*

Ausgehend von dieser Zusammenstellung wurde mit Hilfe von [Wel05] und [All03] ein Entwurf gemacht, wie eine Datenübernahme von Daten der MoUSE in das Active Directory eines Domaincontrollers automatisiert erfolgen kann. Inspiration lieferte dazu neben [Coo03], [Tul04] und [Act05] auch [it-03]. Jedoch waren dies alles grundlegende Aktionen, die auf die Notwendigkeiten der lokalen Umgebung anzupassen waren. Somit entstand das Skript, welches im Anhang B.1 zu finden ist.

Dabei werden anfangs 2 Textdateien eingelesen, wobei von der ersten, mit der Konstanten `LOKNUTZER` verknüpften effektiv nur die letzte Zeile mit Auswirkungen genutzt wird. Diese Zeile soll eine kommaseparierte Liste aller Nutzerkennzeichen enthalten, die in die in `DOMAIN` festgelegte Domäne aufgenommen werden sollen. Somit kann das Skript auch in Subdomänen genutzt werden und dort auch die Auswahl der Nutzer einschränken. Auf Grund der Skriptstruktur werden alle anderen Zeilen der Datei ignoriert. Eine besondere Wirkung hat das Wort „all“, welches allein auf der letzten Zeile stehen sollte und damit alle Nutzerkennzeichen zur Integration vorsieht. Die zweite Datei, welche mit der Konstanten `NUTZER` verknüpft ist, wird zeilenweise eingelesen und ausgewertet und enthält eine Auflistung aller Nutzer mit den entsprechenden Attributen. Erwartet wird eine Textdatei, die pro Zeile einen Nutzer in der Struktur `NKZ : Nachname : Vorname : Titel : Struktur : Mail :`

Telefon : Fax : Raum : Gruppe enthält. Wie ersichtlich, wird der Doppelpunkt als Separator zwischen den Feldern genutzt. Unter „NKZ“ versteht sich das Nutzerkennzeichen, welches dem User-Bestandteil des Kerberosprinzips entspricht und in der MoUSE in der Relation *nutzerkennzeichen* abgelegt ist. „Vor-“ und „Nachname“ sollten sich selbst erklären, Titel ist ein eventueller Titel der Person, welcher in der *person*-Relation zu finden ist. Unter „Struktur“ wäre ein kommaseparierter String der Strukturnamen aus der *struktur*-Relation und einer rekursiven Auswertung des *struktur_parent*-Attributes, vergleichbar der Verwendung der Struktur-Anzeige bei [MoU05], sinnvoll. Das Feld „Mail“ enthält den primären Mailalias des Nutzers aus den Relationen *mailalias* und *mailadr*. Die Felder „Telefon“, „Fax“ und „Raum“ können entsprechend der Datenbereitstellung durch die MoUSE bei Mitarbeitern die entsprechenden Informationen enthalten. Eine Bereitstellung dieser Daten ist angesichts der starken Integration des Active Directorys in das Windows-System sicher sinnvoll, da beispielsweise die Suchfunktion jedes Domänen-Rechners darüber die Suche nach Personen zuläßt und diese mit echtem Mehrwert für den Windowsnutzer eingesetzt werden kann. Somit kann ein Windows-Mailprogramm, wie Outlook, seine Adreßbücher pflegen oder ein Student seinen Dozenten ohne Umweg auf dessen Homepage finden. Die entsprechenden Daten sind in der *person*-Relation abgelegt. Ihre Aktualität sollte sich vor allem im Kontext der VoIP-Einführung erhöhen lassen.

Das Feld „Gruppe“ wird durch das Skript auf eine OU-Hierarchie innerhalb des Active Directorys abgebildet. Dabei ist die Hierarchie innerhalb des Feldes von der Wurzel zum Blatt abgebildet und wird durch „_“ separiert. Also „Studenten_Informatik“ wird auf eine OU „Informatik“ abgebildet, welche in einer OU „Studenten“ liegt, die Bestandteil der Domäne ist. Diese OUs müssen bestehen, werden also nicht durch das angefügte Skript erzeugt. Sicher läßt sich durch einfaches Erweitern der Skriptfunktionalität solches erreichen. Dazu müsste der DN des Zielobjektes zerlegt werden und die letzte existente Hierarchie-Ebene als Container connectet und darin die entsprechend nächste Ebene als OU-Objekt erzeugt werden. Dies gibt das aktuelle Skript nicht her, verhindert damit aber wildes Wachstum des Baumes beispielsweise durch Tippfehler.

Für jede Zeile dieser zweiten Datei findet nun ein Abgleich statt, ob der Nutzer dieser Zeile in die Domäne integriert werden soll. Ist dies der Fall, wird in der Domäne nach dem Nutzer gesucht, um zu entscheiden, ob er neu angelegt oder nur gegebenenfalls aktualisiert werden muss. Ist er neu in der Domain, wird er mit seinen Attributen und einigen Standardwerten angelegt. Dabei ist zu beachten, dass ein Nutzerkennzeichen, Vor- und Nachname erwartet werden und bei Fehlen zu Abbrüchen des Skripts führen, während alle anderen Attribute optional sind. Hier wäre nötigenfalls eine weitere Verbesserung des Skripts vorzunehmen – allerdings sollten diese Informationen eines jeden Nutzers vorhanden sein. Einige Attribute sind auskommentiert, ließen aber beispielsweise Standardwerte eintragen, was mit dem Gedanken einer universitätsübergreifenden Nutzung der Daten in Zukunft interessant werden könnte. Auch wird derzeit ein festes Passwort für alle Accounts eingetragen, was nicht sicher ist und verbessert werden sollte – beispielsweise durch einen Passwortgenerator. Ein Beispiel, wie ein solcher Generator in VBScript aussehen könnte, wird auch auf [faq05] vorgestellt, aber hier wäre beispielsweise auch ein externes Generieren und eine Übergabe in der *NUTZER*-Datei denkbar. Gebraucht wird das Passwort nicht, da ja gegen Kerberos zu authentifizieren ist. Außerdem werden noch einige Attribute gesetzt, die mit der Alterung des Passworts in Zusammenhang stehen, welches bei Windows per default eingestellt ist.

Ein Attribut, welches noch interessant sein könnte, hier aber noch nicht einfluss, ist „account-Expires“, welches ein Ablaufdatum für den Account enthält. Dessen Nutzung setzt aber eine 64Bit-LongInt Zahl als Ablaufdatum voraus, deren Verwendung auf [ADT06] erklärt wird. Solange die Gültigkeit der Kerberos-Einträge beschränkt ist, wirkt sich dies indirekt auch auf die Windows-Einträge aus. Es wäre auch denkbar, zu löschende Einträge im *NUTZER*-File einzuführen und zu verarbeiten. Alternativ ist vorstellbar, während der Abarbeitung die durch die Files für die Domäne vorgesehenen Accounts in einem String zu vermerken und diesen am Ende des Skripts mit dem tatsächlichen Bestand aller Accounts in der Domäne abzugleichen und gegebenenfalls „Überschüssige“ zu löschen. Dies sollte in der Folge der Arbeit genauer betrachtet werden.

Ist hingegen der Nutzer bereits in der Domain gefunden worden, wird als erstes ein Abgleich hinsichtlich des Objektortes gemacht, was beispielsweise bei einem Studiengangwechsel oder dem Übergang vom Studentendasein in ein Mitarbeiterverhältnis denkbar wäre. Stimmt der bisherige Ort nicht mit dem erwarteten überein, wird das Objekt verschoben und an den erwarteten gebracht. Ist das Objekt am erwarteten Ort im Verzeichnis, werden die übergebenen Attribute mit den derzeit im Objekt abgelegten verglichen und gegebenenfalls aktualisiert. Die mit Standardwerten bei der Einrichtung der Nutzer belegten bleiben außen vor, da deren Änderung nicht automatisch erfolgen sollte. Wird ein Attribut aus dem *NUTZER*-File entfernt, wird es im AD mit einem einzelnen Leerzeichen ersetzt, da es keine adäquate Löschfunktion für Objektattribute gibt.

Mittels des angefügten Skripts wurden in der Domäne der Teststellung 10000 Nutzer angelegt, was eine Ausführungsdauer von circa 28 Minuten mit sich brachte. Die spätere Aktualisierung beliebiger dieser 10000 Nutzeraccounts mittels des Skriptes erfolgte in weniger als 5 Minuten je Durchlauf. Also sowohl die einmalige Einrichtung aller Nutzeraccounts als auch ein tägliches oder noch häufigeres Update des Datenbestands ist in vertretbarer Zeit mit dem Skript realisierbar.

5.8 Gruppenarten und Arbeiten mit Gruppen

Der Begriff „Gruppe“ wird im Kontext von Active Directory in zweierlei Hinsicht genutzt, es gibt Sicherheits- und Verteilungsgruppen. Die Zweitgenannten sind einfach und schnell erklärt. Sie dienen ausschließlich der Verteilung von Nachrichten auf mehrere Nutzer, sind also nichts anderes als eine Windowsart von Mailinglisten.

Von den Sicherheitsgruppen hingegen gibt es insgesamt 4 Typen, wovon allerdings jeweils nur eine Untermenge verfügbar ist. Welche Typen nicht nutzbar sind, ist von der Funktionsebene der Domäne und damit auch indirekt von der Funktionsebene der Gesamtstruktur abhängig. Unterschieden werden die 4 Typen **lokale, domainlokale, globale und universelle** Gruppe.

Die *lokale* Gruppe wird ausschließlich in der Domäne verwaltet und ist auch nur in dieser verfügbar. Das heißt, Mitglieder und zu verwaltende Ressourcen müssen in einer Domäne liegen. Dieser Gruppentyp wird mit der Aktivierung des globalen Katalogs in seiner vollen Funktionsbreite durch Heben der Funktionsebene auf mindestens „Windows 2000 pur“ überdeckt, ist also danach nicht mehr verfügbar. An seine Stelle treten die *domainlokalen* Gruppen, welche zur Verwaltung der Ressourcen der eigenen Domain dienen, also ähnlich der

lokalen Gruppe sind. Als Erweiterung kommt hinzu, dass Mitglieder aus anderen vertrauten Domänen kommen können, also Domänen, zu denen eine explizite Vertrauensstellung herrscht beziehungsweise, die in der gleichen Gesamtstruktur und damit in einer impliziten Vertrauensstellung liegen. Die Aktualisierung der Berechtigungen wird dabei vom Infrastrukturmaster der Domäne überwacht und vorgenommen (siehe auch 4.4).

Das Pendant in entgegengesetzter Richtung ist die *globale* Gruppe, welche nur Mitglieder aus der Domäne haben kann, in der sie deklariert wurde, aber eben dieser Gruppe dann wiederum Rechte auf Ressourcen anderer beliebiger Domänen einräumen kann. Grundsätzlich muss der zuweisende Nutzer berechtigt sein, Rechte auf die Zielressourcen zu vergeben, so dass nicht einfach ein Administrator einer fremden Domäne Rechte in einer anderen vergeben kann, in dem er eigene globale Gruppen bevorteilt. Globale Gruppen werden nur innerhalb ihrer Domäne gehalten und repliziert, gehen also nicht in den globalen Katalog ein und sind damit auch nur bei Verfügbarkeit ihrer Domäne erreichbar.

Die letzte Art der Gruppen ist die *universelle*. Dieser Gruppentypus wird im globalen Katalog verwaltet, ist daher auch erst ab der Funktionsebene „Windows 2000 pur“ nutzbar. Daher ist der Aufwand bei diesen Gruppen auch relativ gross, wenn an ihnen Änderungen auftreten. Diese müssen auf alle GC-Server der Gesamtstruktur repliziert werden. Daher wird beispielsweise auf [Gru05] empfohlen, solchen Gruppen nur globale Gruppen als Mitgliedern zuzuweisen, welche dann innerhalb einer Domäne die eigentlichen Mitglieder der universellen Gruppe aus dieser Domain halten. Dadurch verringert sich der Replikationsaufwand bei Auswirkungen, die nur einzelne Nutzer betreffen.

Zusätzlich gibt es auf jedem Rechner noch zusätzliche Gruppen, die nicht dokumentiert sind. Solche Gruppen werden aus den Mitgliedern anderer Gruppen oder aus Tätigkeiten des Nutzers implizit abgeleitet. [KT04] weist in dem thematisch zugehörigen Kapitel 18 solche Gruppen in einer Tabelle aus, zu denen beispielsweise interaktiv angemeldete, per Netzwerk angemeldete oder anonym angemeldete Nutzer zählen. Diesen Gruppen kann man nicht explizit Nutzer zuweisen und auch der Versuch in solche Gruppen, wie „Benutzer“ (also der Sammlung der dem Rechner bekannten Nutzer – üblicherweise eine Menge der lokalen Accounts und Domainnutzer) zusätzliche Gruppen zuzuweisen, hat zwar grundsätzlich funktioniert, aber keinerlei Auswirkungen gezeigt. So war es nicht möglich, dadurch Nutzern anderer Domänen Zugang zum System durch Anmeldung an dieser Domain zu verschaffen.

Eine recht übersichtliche Aufbereitung der Gruppentypen und ihrer Nutzbarkeit durch Verschachtelung findet sich neben oben genannter Quelle unter [Gru05] auch auf [Win00a].

5.9 Strukturierung mittels Organisationseinheiten (OU)

Eine Möglichkeit, die sich neben der hauptsächlich im Auge gehaltenen Strukturierung mittels Domänen und Subdomänen anbietet, ist die Einrichtung von entsprechenden Organisationseinheiten. Neben der Möglichkeit dort Nutzer entsprechend der Zugehörigkeit besser zu strukturieren oder Ressourcen entsprechend der Unternehmensstruktur zu verwalten, lässt sich eine delegierte Administration von Rechnerressourcen vorstellen. Die Administratoren der zentralen Domäne verschieben die Domainaccounts der betroffenen Computer in die Organisationseinheit, weisen einem Domainbenutzer die entsprechenden Rechte zur Erstellung

von Gruppenrichtlinien in dieser Organisationseinheit zu. In der Folge ist der Domänennutzer befähigt, Anpassungen an der Administration der Rechner mit Hilfe von Gruppenrichtlinien durchzuführen.

Dadurch kann beispielsweise ein kompetenter Mitarbeiter Anpassungen vornehmen, die sich eine Arbeitsgruppe zusätzlich zu den Standardeinstellungen wünscht, gesonderte Software verteilen. Dazu ist seinerseits nicht einmal ein direkter Zugriff auf einen Domänencontroller notwendig. Vielmehr kann alles mit der Gruppenrichtlinien-Management-Console bearbeitet werden. Diese steht unter [GPM05] in der aktuellen Version mit Service Pack 1 frei zum Download bereit, wird auch nicht mit der Windows Server 2003 CD ausgeliefert, da ihre Fertigstellung erst kurz nach der Veröffentlichung des Betriebssystems erfolgte. Diese Software ist unter Windows Server 2003 und Windows XP einsetzbar, weshalb ein Domänennutzer diese auch von seinem Arbeitsplatz aus nutzen kann, um Anpassungen vorzunehmen. Zusätzlich geht die administrative Hoheit des Universitätsrechenzentrums nicht verloren, da ein solcher Nutzer nur neue GPOs (Gruppenrichtlinieobjekte) erzeugen kann, die zusätzlich zu den bereits Vorhandenen angewandt werden. Über die Reihenfolge der Anwendung wurde bereits in 4.7 gesprochen. Diese wirkt sich natürlich auch hier aus und es sei erwähnt, dass beispielsweise das vererbte Default-GPO der Domäne in der OU zuerst angewandt wird, also das neu hinzugefügte spezialisierende Wirkung hat.

Um dieses Werkzeug zu nutzen, muss der Nutzer das MSI-Paket nur auf seinem Arbeitsplatz installieren, was dank der Bereitstellungsform auch durch die zentrale Administration stattfinden kann. Danach findet sich unter `%SYSTEMROOT%\system32` eine MMC-Verknüpfung namens `gpmc.msc`, über die das Tool geöffnet werden kann. Um das Werkzeug dann mit dem mittels externem Kerberos authentifizierten Account nutzen zu können, muss noch die Vertrauensüberprüfung in den Optionen des MMC-SnapIns deaktiviert werden. Danach ist eine Verbindung mit der Domäne möglich und der Einrichtung eines neuen GPOs in der zugewiesenen OU steht nichts mehr im Wege.

Dabei stellte sich im Rahmen der Tests heraus, dass die Nutzung von Einstellungsmöglichkeiten absolut unproblematisch von Statten ging, während die Softwareinstallation Schwierigkeiten bereitete. Ursache der Schwierigkeiten ist die adäquate Bereitstellung der Installationsquelle. So muss ein Computer, der durch ein GPO eine zu installierende Software zugewiesen bekommt, auf die Netzwerkfreigabe zugreifen können, auf welcher die Paketdatei abgelegt ist. Einem im Active Directory angelegten Freigegebenen Ordner können in „Active Directory Benutzer und Computer“ unter „Sicherheit“ Berechtigungen zugeordnet werden, die sich allerdings nur auf die Freigabeberechtigungen auswirken. Hat der Rechner auf den eigentlichen Ordner keinen Zugriff, wird ihm dieser auch trotz der für die Freigabe gesetzten Rechte verwehrt. Zur Rechteverwaltung des Filesystems kann man den Ordner aber im Explorer öffnen lassen und dort unter Eigenschaften die Berechtigungen des Filesystems anpassen.

Trotz dieser Umstände, die sich durch geschickte Rechteverwaltung umgehen lassen, kann somit für einen Teil der Anwendungsfälle eine sinnvolle Lösung bereitgestellt werden, mit der den Wünschen mancher Struktureinheit sicherlich Genüge getan werden kann.

5.10 Firewall-Aspekte

Auf Grund seiner Verbreitung ist das Windows-Betriebssystem oftmals Ziel vielseitiger Angriffe aus dem Inter– aber auch Intranet. Viren, Würmer und Hacking-Skripte sind nur eine kleine Auswahl der üblichen Gefahren. Auch fehlerkonfigurierte Klienten können zu Problemen führen. Daher werden Server üblicherweise nur hinter einer Firewall oder in einer so genannten DMZ betrieben. Für den Betrieb bleibt somit allerdings die Frage, welcher Netzwerkverkehr zum Server durchgelassen werden muss, um den Betrieb des eigentlichen Dienstes nicht zu beeinflussen.

Ein Windows Domaincontroller, der ein Active Directory bereitstellt, bietet verschiedene Dienste, auf die der Zugriff ermöglicht werden muss, um den erfolgreichen Betrieb zu gewährleisten. Das sind einerseits das Kerberos-Protokoll, aber auch der LDAP-Zugang zum Verzeichnis. Andererseits muss auch der globale Katalog bedacht werden und die windows-typischen RPC-Systeme für CIFS und SMB für Drucker- und Netzfreigaben. Da beispielsweise die Replikation, aber auch die Softwareinstallation mittels dieser RPC-Systeme funktioniert, sind diese auch unerlässlich für den regulären Betrieb. Somit sind bereits folgende Ports erreichbar zu halten:

Port	Protokoll	Anwendung
Notwendige Dienste für einen Domaincontroller		
88	tcp/udp	Kerberos
389	tcp	LDAP-Standardport
464	tcp/udp	Kerberos V5 Passwortdaemon
636	tcp	LDAP (SSL-verschlüsselt)
3268	tcp	globaler Katalog (LDAP-Protokoll)
3269	tcp	globaler Katalog (SSL-verschlüsselt)
Um WinNT-, Win9x-, WinME-Clients bedienen und die Windows-Netzwerkumgebung bereitstellen zu können		
135	tcp/udp	RPC-Portmapper
137	tcp/udp	NetBIOS Name Server
138	udp	NetBIOS Datagramm
139	tcp	NetBIOS Session Services
445	tcp/udp	SMB-Protokoll (Freigaben)
Weiterhin sinnvoller Dienst		
3389	tcp	Remote Desktop Protocol

Zusätzlich ist auf Grund der dynamischen Portzuteilung der RPC-Dienste durch den RPC Portmapper keine sinnvolle Porteinschränkung jenseits des privilegierten Portbereiches möglich.

Um diesen Umstand gerecht werden zu können, ist es mit Hilfe von Manipulationen an der Registry möglich, alle RPC-Aufrufe auf einen fest vereinbarten Port abzubilden, wodurch nur noch dieser erreichbar sein muss, um die volle Funktionalität zu gewährleisten. Dazu muss der entsprechend gewählte Port unter *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NTDS\Parameters* als *DWORD*-Wert namens *TCP/IPPort* eingetragen werden. In der Folge wird der RPC-Portmapper auf Port 135 nur noch diesen einen Port als Konnektivitätsmöglichkeit melden. Entsprechend muss die obige Liste der

Ports für die Firewall nur noch um diesen einen Port ergänzt werden.

Alternativ kann auch die Windows-Firewall beeinflusst werden, einen eingeschränkten Bereich zur weiterhin dynamischen Nutzung freizugeben. Dazu werden auch Registry-Keys erzeugt, die die Firewall entsprechend konfigurieren. Genauer dazu und Weiterführendes kann im Kapitel „Active Directory und Firewalls“ bei [KT04] nachgelesen werden.

In der Folge des Kapitels finden sich ebenfalls Ausführungen über eine weitere Sicherungsmöglichkeit mittels VPN beziehungsweise mittels PPP sowie Zertifikaten. Ersteres kann nur zur Replikation und zum Datenverkehr zwischen Domaincontrollern genutzt werden, ist Kerberosbasiert und daher nicht für den Beitritt eines DCs in eine Domäne realisierbar. Letzteres zieht die Notwendigkeit zum Aufbau einer Zertifikatstruktur nach sich, kann dann allerdings zur Sicherung jeglichen Traffics zwischen Domaincontrollern genutzt werden. Zur Einführung in die beiden Themen soll an dieser Stelle auf [KT04] verwiesen sein, sprengt aber in voller Breite selbst deren Rahmen und wäre hier weit ab vom Themenkern.

5.11 AD als LDAP-Ersatz

Da das Active Directory in seiner zentralen Schnittstelle auf den offenen Standard des Lightweight Directory Access Protocols (LDAPv3) aufbaut, kommt im Zusammenhang der Integration eines solchen als Dienst schnell die Frage auf, in wie weit das Verzeichnis auch als Verzeichnisdienst beispielsweise einen OpenLDAP-Server ersetzen kann. Derzeit existiert ein einzelner PC, der als LDAP-Server für Recherchezwecke zum Beispiel von E-Mailprogrammen dient.

Das bedeutet, dass ein wesentlicher Aspekt in dieser Hinsicht die Zugriffssteuerung ist. Standardmäßig kann auf ein Active Directory eines Windows Server 2003 durch einen kerberos-authentifizierten Nutzer mit entsprechenden Domainbenutzer-Rechten zugegriffen werden. Die Berechtigungssteuerung kann über das GUI des MMCs „ADSIEdit“ erfolgen, welches erst nach der Installation der Support Tools von der Install-CD verfügbar ist. Dabei ist zu beachten, dass es eine Sondergruppe „Jeder“ gibt, welche alle interaktiv oder über das Netzwerk angemeldeten Nutzer zusammenfasst. Somit ist zwar jeder irgendwie authentifizierte Nutzer befähigt, dieser Rolle gerecht zu werden, aber eben niemand, der über keinen gültigen Account in der Domäne verfügt. Da die Authentifizierung größtenteils sowieso auf Kerberos aufsetzt, ergibt sich aus der Rechtevergabe an die Gruppe „Jeder“ meist kein Vorteil.

Die im Windowsumfeld wesentlich seltener genutzte Sondergruppe „Anonymous-Anmeldung“ erstellt hingegen die Möglichkeit, auch nichtauthentifizierte Nutzern den Zugriff auf bestimmte Aspekte des Verzeichnisses zu gewähren. Somit kann etwa ein Mailprogramm aus einem Nutzerkennzeichen, eine Mailadresse auflösen oder einen vollen Namen ermitteln. Das ausschließliche Setzen dieser Rechte hat im Windows 2000 noch ausgereicht, um einen entsprechenden Zugriff zu gewähren. Aus Sicherheitsgründen ist dies im Windows 2003 unterbunden worden und muss bei Wunsch durch Anpassungen der Registry explizit erlaubt werden. Einen ausführlichen Hinweis zum sinnvollen Setzen von Rechten, um Mailprogramme mittels LDAP an das Active Directory zu binden, findet der Leser unter [Ano04]. Die Anpassungen, die seitens der Registry notwendig sind, um anonyme Zugriffe grundsetzlich zu erlauben, finden sich auf [Ano05].

Ein weiterer wesentlicher Hinweis in diesem Zusammenhang, der sich auch auf oben genannter Webseite findet, ist die Notwendigkeit bei anonymen Zugriffen auf die globalen

Kataloge des GC-Servers zuzugreifen. Somit ist einerseits nur eine Auswahl der Domain-controller in einer Gesamtstruktur für anonyme Zugriffe zu nutzen, andererseits beim Zugriff dem Clienten ein expliziter Port, nämlich 3268/tcp, mitzuteilen.

Ein Versuch mit einem Thunderbird (siehe [Thu05]) Mailclients hat gezeigt, dass eine unkomplizierte Anbindung eines Active Directorys an dieses Mailprogramm möglich ist, wodurch die LDAP-Schnittstelle als Adressbuch genutzt werden kann. Im folgenden Bild sehen Sie die Einstellungen, welche bekannten Einstellungen zu anderen Programmen entsprechen - es sei nur auch nochmals auf die Anpassung der Portnummer hingewiesen.



Abbildung 5.5: AD als Adressbuch eines Mailprogramms

5.12 Alternativen zu Active Directory

Grundsätzlich stellt Active Directory einen Verzeichnisdienst nach X.500-Standard mit einer laut Microsoft standardkonformen LDAP-Schnittstelle dar. Somit bietet der Dienst einerseits die Möglichkeit, Systeme anzubinden, die einen LDAP-konformen Verzeichnisdienst erwarten, andererseits ist AD nicht der einzige Vertreter solcher. Somit stellt sich relativ schnell die Frage, ob Active Directory alternativenlos dasteht.

Kurz um läßt sich vorab zusammenfassen, dass Active Directory wegen seiner starken Integration in das Windows Betriebssystem und der Vielfalt seiner zur Sicherung der Abwärtskompatibilität bereitgestellten Schnittstellen schon ein paar Alleinstellungsmerkmale aufweist, die ein Konkurrenzprodukt nur schwerlich kompensieren kann.

Alternativen sollten alle Dienste ersetzen, die hier im Zusammenhang mit Active Directory genannt wurden, also einen Verzeichnisdienst mindestens mit LDAP-Schnittstelle bereitstellen, Authentifizierung auf Kerberos-Basis, möglichst auch per NTLM, ermöglichen, eine entsprechend konfigurierte DNS-Umgebung und soweit gewünscht, was in den meisten Anwendungsfällen so sein wird, einen Datei- und/oder Druckserver bereitstellen. Alle diese Dienste gibt es natürlich auch als freie Entwicklungen oder auch von anderen Herstellern. Allerdings erwartet ein Windowsclient diese in einer ziemlich „kompakten“ Umgebung.

Wie bereits angesprochen, kann die DNS-Umgebung auch durch den Einsatz MS-fremder Server sogar mit statischem DNS geschaffen werden, wodurch dies im Rahmen der Alternativenbetrachtung außen vor bleibt. Die Kerberos-Authentifizierung kann, wie ebenfalls aufgezeigt, durch beispielsweise MIT- oder Heimdal-Kerberos ersetzt werden. Bei entsprechender DNS-Konfiguration wird ein AD-Client diesen externen Kerberos-Server auch auffindig machen und nativ nutzen. Ungeklärt bleibt im Rahmen dieser Arbeit die Integration verschiedener AD-Dienste in eine solche Umgebung, welche mit der automatischen beziehungsweise internen Vergabe von Prinzipal-Passwörtern einhergeht. Schwieriger wird bereits die Bereitstellung einer stets mit Kerberos korrelierenden NTLM-Authentifizierung. Einen entsprechenden Ansatz bietet Samba, welches primär auch als Datei- und Druckserver für Windows-Clients gedacht ist. Einen Verzeichnisdienst auf LDAP-Basis können mehrere Implementierungen bereitstellen, zum Beispiel auch OpenLDAP.

Es gibt allerdings zwei Entwicklungen, die näherungsweise sich dem Ziel verschrieben haben, eine Unix/Linux-Umgebung für Windows-Clients nutzbar zu machen, auf die hier kurz eingegangen werden soll.

5.12.1 PADL/XAD

XAD ist ein kommerzielles Produkt, welches von der Firma PADL angeboten wird und verspricht, sämtliche durch ein AD bereitgestellten Dienste zu ersetzen, sich dabei allerdings als Identity Management System versteht und darauf beschränkt. So wird mittels Heimdal, Bind, Samba, SASL und OpenLDAP sowie weiteren Diensten eine Umgebung konstruiert, die der eines Windows Active Directory nahe kommen soll. Verfügbar ist das Produkt nach Herstellerangaben nur für SuSE Linux 9.1, den SuSE Linux Enterprise Server 9 auf verschiedenen Hardware-Plattformen und das Red Hat Enterprise Linux 4 System für x86.

Auf Grund dessen, dass es ein kommerzielles Produkt ist, konnte im Rahmen der Arbeit keine Verifikation der Funktionalität oder weitergehende Tests mit einer entsprechenden Umgebung gemacht werden.

Der entscheidende Vorteil, den das Produkt mitbringt, ist die einheitliche Administrationschnittstelle, die für die notwendige Verknüpfung aller Dienste sorgt. Ansonsten enthält das Paket keinerlei Komponenten, die nicht auch frei verfügbar wären und somit im Rahmen einer anderen Arbeit als Kombination getestet werden könnten.

Soweit aus den Ausführungen der Produktwebseite unter [XAD05] ersichtlich, arbeitet XAD mit einem dynamischen DNS-System aber auch einem gepatchten NTP-Server, der signierte Antworten erzeugt, wie sie von Windows-Clients erwartet werden. Weiterhin ist das Heimdal-Kerberos entsprechenden Anforderungen hinsichtlich gepatcht, so dass ein Ersatz des produkteigenen durch einen bestehenden ausgeschlossen wird.

Eine Frage, deren Antwort allerdings allein aus dem Studium der Produktinformationen nicht hervorgeht, bleibt sowohl hier als auch bei einer möglichen Konstruktion auf eigenen Diensten offen. *Wie findet das ziemlich komplexe und umfangreiche Schema eines ADs Einzug in die LDAP-Umgebung einer externen Konstruktion und wie kann der steten Erweiterung durch Microsoft oder externen Produkten Rechenschaft getragen werden?* Vermutlich wird eine entsprechende Konfiguration in den RPMs der Produktdistribution mitgeliefert, allerdings bleibt offen, wie umfangreich diese ist, welche Windows-Server-Version dafür Pate stand und ob eine dynamische Erweiterung durch Produktinstallationen möglich ist. Auch

werden Einschränkungen bezüglich der Interoperabilität mit „echten“ Active Directory Domänen eingeräumt – etwa kann eine XAD-Domäne nicht Mitglied eines AD-Baumes werden und eine Childbeziehung zu einer solchen kann ebenfalls nicht aufgebaut werden, nur Cross-Realm-Trusts zu solchen sind möglich. Datei- und Druckserverdienste werden trotz der Integration von Samba nicht angeboten, da von Samba nur die CIFS-Komponenten integriert wurden, um den RPC-Dienst anzubieten. Es wird aber darauf hingewiesen, dass Samba an XAD gebunden werden kann, um dessen Dienste zu nutzen. Ebenso ist eine Anbindung eines OpenAFS-Dienstes an XAD vorgesehen und realisierbar.

Teile der Anpassungen, die PADL an den genutzten Produkten vorgenommen hat, werden auf der Produktwebseite als kostenloser Download bereitgestellt.

Trotz allem bleibt es also zu bezweifeln, dass trotz der komplexen Struktur des Systems und der grundlegenden Bereitstellung aller notwendigen Komponenten ein vollständiger nativer Windows-AD-Ersatz durch XAD zu erreichen ist. Spätestens bei der Planung des Einsatzes von Applikationen, die etwa auf die MAPI-Schnittstelle zurückgreifen, sind die Grenzen des Verfahrens erreicht.

5.12.2 SAMBA

Anders als bei dem gerade vorgestellten Identity Werkzeug liegt das primäre Anliegen der Samba-Entwicklung in der Bereitstellung von Datei- und Druckdiensten für Windows-Clienten durch Unix/Linux-Server. Samba ist eine OpenSource-Entwicklung, die auf [Sam05] beheimatet ist. Auf Grund der Zielstellung liegen auch die programmtechnischen Schwerpunkte ganz anders, als bei XAD.

Samba stellt einen SMB- und CIFS-Server bereit, der von Windows Clienten und Servern als File- oder Druckserver genutzt werden kann. Im Rahmen dieses Anliegens ist es natürlich auch notwendig, Nutzer zu authentifizieren, um einer Authorisierung sinnvoll gerecht werden zu können. Doch hier sieht sich Samba viel mehr als Dienstanutzer, denn als Server. Man kann Samba an verschiedene Authentifizierungsmechanismen binden, nicht zuletzt an Kerberos oder PAM. Somit ist auch die gerade angesprochene Anbindung an XAD, aber auch an jeden anderen externen Kerberos-Dienst möglich.

Samba ist auch in der Lage, eine NT4.0-Domain zu simulieren und als Domaincontroller einer solchen zu dienen. Dabei kann Samba als PDC, BDC oder Domain Member Server arbeiten, wodurch auch eine Anbindung an eine bestehende Active Directory Domäne möglich ist, solange sich in dieser ein PDC-Emulator befindet, also solange diese im „Windows 2000 Mixed-“ beziehungsweise im „Windows 2003 Interims Modus“ steht. Samba ist hingegen nicht in der Lage AD Clienten zu bedienen, da weder ein Verzeichnis bereitgestellt wird, noch die entsprechenden Schnittstellen vorhanden sind. Ein Samba-Server kann also nicht die Funktionalität eines Active Directory Servers ersetzen.

5.12.3 Keine Domäne

Natürlich ist der Betrieb auch ohne eine Domäne denkbar, ähnlich des Modells, welches derzeit im Einsatz ist. Auch in diesem Falle ist eine windowsnative Authentifizierung gegen ein externes Kerberos-System möglich, in dem die oben beschriebenen Anpassungen der

Registry des Clientensystems durchgeführt werden. Da der Rechner dann allerdings nicht Mitglied einer Domäne ist, müssen auf allen Clienten-PCs jeweils alle Nutzereinträge gepflegt werden, zumindest beschränkt auf den Nutzerkreis derer, die sich am entsprechenden Rechner authentifizieren können sollen. Dies bedeutet einen erheblichen Mehraufwand und sicher auch hardwareseitigen Mehrbedarf (Festplattenkapazität für jedes Nutzerprofil auf jedem einzelnen Clienten), der in einer Umgebung von mehreren tausend Nutzern inakzeptabel erscheint.

Außerdem wird dadurch dem Ziel der Arbeit nicht Rechenschaft getragen, da somit vorerst weiterhin kein Verzeichnisdienst mit den Daten aller Nutzer bereitgestellt wird, wie er durch eine zentrale GroupWare-Lösung und auch andere Anwendungen genutzt werden kann. Noch weniger wird man dem Bedarf mancher Anwendungen nach einem nativen Active Directory gerecht.

Grundsätzlich läßt sich wohl zusammenfassen, dass viele Aspekte eines Active Directories durch andere Systeme ersetzbar sind, ein vollständiger Ersatz allerdings nur schwerlich zu erreichen sein wird, um Windows-Clienten in diesem ohne größere Anpassungen nativ betreiben zu können.

Dem Wunsch zur Integration der bereits vorhandenen Inzellösungen in das Identity Management des URZs beziehungsweise der Eingliederung in einen Domänenbaum, der eine zentral als Dienst betriebene AD-Domäne als Wurzel hat, wird durch keine der Alternativen ausreichend nachgekommen.

6 Fazit

6.1 Allgemeines

Ausgehend von dem im vorherigen Kapitel beschriebenen Erfahrungen läßt sich kurz bereits anfänglich sagen, dass es leider keine Lösung im Sinne einer Beschreibung gibt, wie die Systemumgebung eingerichtet werden muss, um ein Active Directory als Dienst anzubieten wie es in der Intension der Aufgabenstellung liegt.

Wie im vorherigen Kapitel angesprochen, sind viele Punkte des Vorhabens ein Active Directory als Dienst ohne massive Änderungen der gegebenen Struktur erfüllbar, aber leider nicht alle.

Wie in der in 6.2 folgenden Diskussion klar werden soll, gibt es mehrere Varianten sich dem Ziel zum Einsatz eines Active Directories zu nähern, wobei keine nachteilsfrei ist und somit es in diesem Kapitel um die Unterbreitung einer Diskussionsgrundlage geht, welche dann im URZ zu einer politischen Entscheidung führen soll, ob und wenn ja auf welchem Weg das Ziel verfolgt werden soll.

Es ist möglich, eine Einbindung des DNS in die vorhandene BIND-Konfiguration vorzunehmen, sogar auf die Notwendigkeit eines dynamischen Updates zu verzichten. Grundlage dafür sind die Anmerkungen aus 5.5 und die Notwendigkeit durch eine konsistente Konfiguration, die Abbildung zwischen Diensten, Servern und Clienten sowie den jeweiligen IPs sicher zu stellen. Auf Grund dessen konnten auch keinerlei Einschränkungen hinsichtlich des Einsatzes von DHCP bisher ausfindig gemacht werden. Die Dienste werden grundsätzlich auf Servernamen abgebildet und solange die Abbildung zwischen Rechnernamen und IP-Nummern konsistent zur Vergabe mittels DHCP bleibt, sprachen auch die Versuche nicht gegen einen Einsatz von dynamisch zugewiesenen IPs. Besonderes Augenmerk sollte dabei aber auch den „forwarders“- und „NS“-Einträgen in der BIND-Konfiguration gewidmet werden.

Es ist ebenfalls möglich, eine Windows Active Directory Domain an einen externen Kerberos-Dienst zu binden, solange in dieser Domäne für die Bereitstellung entsprechend zugeordneter Nutzerobjekte Sorge getragen wird. Dies funktioniert dank Cross-Realm-Trusts beziehungsweise, wie es bei Windows genannt wird, Vertrauensstellungen und einigen wenigen Einstellungen auf den Clientensystemen mit Windows XP, Windows 2000 Professional und Windows 2003 als Betriebssystemen.

Ebenso funktioniert durch die Einrichtung expliziter oder impliziter Vertrauensstellungen der domänenübergreifende Zugriff auf Ressourcen fremder Dienstanbieter. Implizite Vertrauensstellungen entstehen beispielsweise, wenn eine Domain in eine existierende Gesamtstruktur eingegliedert wird, explizite werden im MMC-SnapIn „Active Directory Domänen und Vertrauensstellungen“ eingerichtet, so auch die zu einem externen Kerberos-Realm.

Hingegen gibt es, wie bereits erwähnt, Einschränkungen beim Einsatz von Heimdal-Kerbe-

ros als externen KDC. Darauf soll in der Folge des Kapitels noch eingegangen werden.

Weiterhin stellen Gruppenrichtlinien und die Delegierung der Einrichtung solcher an einfache Nutzeraccounts eine gute, hierarchische Möglichkeit der Rechnerverwaltung und auch Softwareverteilung zumindest einiger grundlegender Software, wie etwa des OpenAFS-Clients, welche dadurch unabhängig von ihrem eigenen Dienst wird. Auch läßt sich vorstellen, kleineren Struktureinheiten damit einen guten Mittelweg zur Systemadministration anbieten zu können, indem die Clientenrechner grundlegend in der Verwaltung des URZ verbleiben, aber zur spezialisierten Anpassung an Anforderungen der Struktureinheit an einen versierten Nutzer dieser vermittelt werden können. Dabei ist die Einrichtung einer OU denkbar, in der die Rechneraccounts abgelegt werden und in der ein Nutzer der Struktureinheit die entsprechenden Rechte zur Verwaltung erhält. Auch dies soll in der Folge noch angesprochen werden.

Ebenso kann in Hinblick auf den Einsatz eines Active Directory als Verzeichnisdienst mit LDAP-Schnittstelle ein positives Votum abgegeben werden. Einzig ist der Einsatz dahingehend zu bedenken, dass eine Erweiterung der Attribute eines Objekttypes die Erweiterung des Active Directory Schemas nach sich zieht. Dazu sei auf 6.1.1 verwiesen. Außerdem muss bedacht werden, dass eben nicht der Standardport für LDAP genutzt wird, sondern der des globalen Katalogs, was aber gegebenenfalls auch durch den Einsatz einer geschickten Firewallkonfiguration umgangen werden kann. So wäre denkbar, dass für einen Alias „ldap.tu-chemnitz.de“ eine transparente Portumleitung von Port 389 auf Port 3268 erfolgt.

6.1.1 Anmerkungen zum Schema

Das Schema ist eine zentrale Komponente jedes Verzeichnisdienstes, so auch beim Active Directory. Grundlegend muss bemerkt werden, dass ein Schema immer für die Gesamtstruktur Anwendung findet und wie in 4.4 erwähnt, nur über den Schemamaster der Gesamtstruktur geändert werden kann. Somit sollte einerseits die Rechtevergabe der Gruppe „Schema-Admins“ sinnvoll gering gehalten werden, andererseits stets bedacht werden, dass Anpassungen des Schemas Auswirkungen auf alle betroffenen Objekte der Gesamtstruktur haben und natürlich einen weitverzweigten Aufwand in der Replikation auf alle Domaincontroller der Gesamtstruktur verursachen.

Dieser Aspekt sollte einerseits bedacht sein, sollte das AD als Ersatz eines bisherig autonom betriebenen LDAP-Dienstes eingesetzt werden wollen, andererseits wirkt er sich bei der Installation von schemaerweiternden Applikationen¹ aus. Darauf soll in der noch folgenden Strukturdebatte ebenfalls eingegangen werden.

Das Schema an sich kann mit Hilfe des MMC-SnapIns „ADSI Edit“ bearbeitet werden. Dieses SnapIn ist Bestandteil der Support Tools, muss also explizit von der Installations-CD nachinstalliert werden.

6.1.2 Alternativen zu Active Directory

Ausgehend von der tiefen Verwurzelung des AD-Dienstes in das Windows Betriebssystem und den damit verbundenen impliziten Architekturvorstellungen eines Windows Betriebs-

¹beispielhaft sei hier MS Exchange genannt

systems, kann wohl kurz und knapp gesagt werden, dass derzeit keine adäquate Alternative bereitgestellt werden kann, um eine Windows Domäne zu betreiben.

Natürlich kann keine endgültige Aussage über die Möglichkeiten des PADL/XAD-Systems (siehe [XAD05]) getroffen werden, doch einige Aspekte sind ja bereits in der Diskussion genannt worden, die zumindest mit heutigem Erkenntnisstand fraglich bleiben.

Weitere echte Alternativ-Systeme konnten nicht ausfindig gemacht werden, die den Umfang eines nativen Active Directorys ersetzen könnten.

6.2 Strukturdebatte

Ausgehend vom jetzigen Erkenntnisstand sind zwei Arten einer Umsetzung denkbar, die jeweils ihre Randbedingungen mitbringen und ebenso ausgehend von der vorhandenen Struktur ihre Vor- und Nachteile aufweisen. Grundlegend sind diese Vor- und Nachteile jeweils abzuwägen und vorweg zu sagen, dass eine Entscheidung zum Aufbau eines Active Directory Dienstes teils weitreichende Folgen haben kann und daher eine politische Entscheidung mit vielseitigen Folgen ist.

Daher seien die konsequent folgenden Varianten hier aufgezeigt und gegenübergestellt und ein Vorschlag zum sinnvollen Einstieg gebracht. Doch erst noch einmal etwas zur Namenspolitik.

6.2.1 Namensraum

Da Windows die Hoheit über einen zugewiesenen Namensraum voraussetzt, um seine Serverdienste anzubieten, haben eines beide Varianten gemein. Allgemein sollte vor Umsetzung eines Ansatzes eine klare Namensraumstruktur geschaffen werden. Wie in 5.4 erläutert, wäre eine Strukturierung sinnvoll, die eine saubere Trennung der Windows-AD-Dienste von sonstige Netzdiensten sichert. Wie ebenfalls bereits angesprochen, ist eine Struktur denkbar, in der einerseits Windows-Gesamtstrukturen auch durch implizite Vertrauensstellungen auf Basis ihres DNS-Suffixes wachsen können, wie oben beispielhaft „ad.tu-chemnitz.de“ oder „windows.tu-chemnitz.de“ für die zentrale Root-Domain genannt, andererseits sind Individualstrukturen im vorhandenen DNS-Baum denkbar, wie etwa „ad.mb.tu-chemnitz.de“ oder „windows.informatik.tu-chemnitz.de“.

Zu beachten ist, dass innerhalb des Namensraums einer Domain jeweils nur die Domaincontroller liegen müssen, da bei Clients auch bei Anbindung an eine Domäne der DNS-Suffix von dieser abgekoppelt werden kann. Bei DCs ist dies nicht der Fall.

Dieser Punkt einer Namenspolicy sollte die Grundlage jeglichen weiteren Vorgehens bilden. Wird sich für die Bildung einer einzelnen Gesamtstruktur mit angegliederten Subdomänen entschieden, sollte dies in einem Raum á la „ad.tu-chemnitz.de“ geschehen, bei Insellösungen beziehungsweise dem parallelen Wachstum vieler Domain-Bäume ist eher die Variante der „ad.struktureinheit.tu-chemnitz.de“ zu empfehlen, um dann implizite Vertrauensstellungen und damit verbundene Probleme bei der Domäneinrichtung zu unterbinden.

6.2.2 Aufbau einer zentralen AD-Gesamtstruktur mit Subdomänen

Ausgehend von der Interpretation der Aufgabe entsteht in der ersten Variante im Universitätsrechenzentrum eine zentrale Windows Struktur die als globale Active Directory Wurzel aller Windows-Domänen dient, die im Campusnetz der TU Chemnitz betrieben werden sollen. In das Verzeichnis dieser Domain werden alle Nutzerkennzeichen des URZs automatisiert mittels des oben entworfenen Skripts und der Datenbereitstellung durch die MoUSE in ein entsprechend strukturiertes Textfile integriert.

Struktureinheiten, die eine eigene Windows-Domain betreiben wollen, integrieren diese als Subdomain in die vorhandene Gesamtstruktur, so dass ein Domänenbaum entsteht, dessen Wurzel die durch das URZ betriebene Domain darstellt. Durch diese Integration in die vorhandene Windows-Struktur entfalten sich über alle beteiligten Domänen Vertrauensstellungen, die einfach aus der Integration implizit entstehen. Zur Verkürzung des Trust-Paths bei der Zusammenarbeit mehrere Substrukturen können explizite Vertrauensstellungen diese Struktur noch untermauern.

Grafisch ist diese Variante nochmal in Abbildung 6.1 dargestellt, wobei die gestrichelten Linien, die impliziten Vertrauensstellungen andeuten.

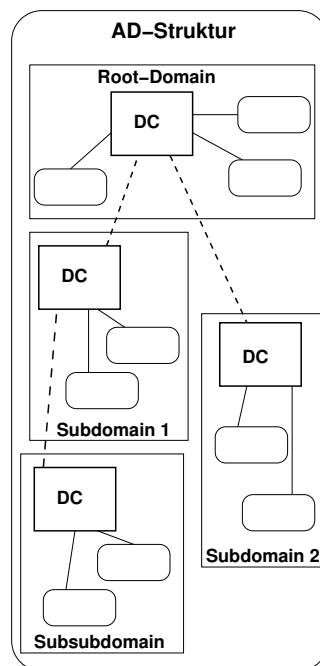


Abbildung 6.1: Active Directory Baum mit Subdomänen

Um allerdings, die Bereitstellung der Nutzeraccounts zentral in der Wurzel zufriedenstellend durchführen zu können, wird es notwendig, Erweiterungen am bestehenden externen Kerberos-System vorzunehmen. Dabei sind zwei Wege denkbar, deren Realisierung einer genaueren Untersuchung bedürfen. Der eine wäre eine hausinterne Erweiterung des Heimdal-Servers, mit den Fähigkeiten der TGS-Request-Abarbeitung analog zum MIT-System, also der Zerlegung von Requests in deren Realm-Bestandteile und einer adäquaten Weiterver-

mittlung aller TGS-Requests aus dem Windowsbaum an die Wurzeldomain der Windowsstruktur. Alternativ ist der Aufbau eines MIT-Kerberos als parallelen Dienst denkbar, wobei der Datenabgleich zwischen Heimdal und MIT automatisiert mittels der Werkzeuge beider Systeme erfolgen sollte. Es bliebe zu testen, ob ein Abgleich mittels Export der Kerberos-Datenbank im Heimdal und Import im MIT erfolgen kann, da eine Kommunikation beider Propagation-Daemons nicht funktioniert. Damit bliebe immer eine Verzögerung der Wirksamkeit von Änderungen bis zum nächsten Abgleich. Außerdem muss untersucht werden, ob die beiden Systeme nebeneinander als Server desselben Realms koexistieren können. In dem Falle der „manuellen“ Abgleichs via Ex- und Import muss sichergestellt werden, dass über die Windows-Clients keine Passwortänderung stattfindet, da diese auf den MIT ge- und beim nächsten Abgleich überschrieben würde.

Ebenso ist ein wesentlicher Faktor bei diesen Gedanken in dieser Arbeit nur beiläufig betrachtet worden, da es den Rahmen sprengte, und so bliebe es zu untersuchen, welche Auswirkungen entstünden und welche Bedingungen notwendig wären, um Schemata-Erweiterungen oder Änderungen durchzuführen. So ist bereits der Wunsch einer Struktureinheit einen Exchange-Server zu betreiben, damit verbunden, gesamtstrukturweit ein neues Schema durchzusetzen, was einerseits zum Zeitpunkt der Installation der Applikation der Synchronisation bedarf und danach der eventuell notwendigen Anpassungen durch alle Domainadmins.

Ergänzend sei zu erwähnen, dass ein Exchange-Dienst nur einmalig in einer Gesamtstruktur sein darf und dort nur einer Administrationsinstanz unterliegt. Somit sollte dieser Dienst in einer Gesamtstruktur zentral angeboten werden. Allerdings sind individuelle Anpassungen des Systems an Anforderungen der Struktureinheiten in der Folge nur noch in Absprache mit allen Nutzern und durch deren Zustimmung zufriedenstellend durchzusetzen. Aber Exchange dient im Rahmen dieser Arbeit nur als populäres Beispiel für viele andere Anwendungen, die ähnliche Anforderungen stellen. Im Extremfall bliebe es auch bei dieser Konstellation nicht aus, Insellösungen in einem eigenen Namensraum zu schaffen, um derartigen Anforderungen beziehungsweise Wünschen gerecht zu werden.

Als weiterer Aspekt, der einschränkend hinzukommt, sei aufgeführt, dass die Integration weiterer Domänen jeweils den Eingriff eines Domain-Administrators der hierarchisch übergeordneten Domäne erfordert. Dies ist ein einmaliger Aspekt im Rahmen der Einrichtung einer Domäne, sollte aber bedacht werden und wird daher in der Folge auch nochmals erwähnt.

Auch sollte bedacht sein, dass eine Einschränkung des Nutzerkreises eines beliebigen Windowsrechners in der Gesamtstruktur nur mehr physisch möglich ist, da sich auf dem Trust-Path bei der Anmeldung durch die zentrale Bereitstellung der Nutzeraccounts immer eine gültige Nutzerinstanz findet.

Vorteilhaft bei der genannten Konstellation ist allerdings diese einmalige zentrale Bereitstellung und Pflege der Nutzeraccounts, wobei diese dann auch noch durch einen gering gehaltenen Personenkreis mit den entsprechenden Privilegien erfolgt. Die Domain-Administratoren der angegliederten Substrukturen brauchen sich nur noch um die Verwaltung und Rechtevergabe der in ihren Domänen bereitgestellten Ressourcen kümmern. Auch wird dadurch sicher der Aktualität der entsprechenden Einträge besser Rechenschaft getragen.

6.2.3 Aufbau von individuellen Domänen/Gesamtstrukturen

Eine Alternative zum gerade vorgestellten Konzept ist die Variante, alle Windows-Domänen in jeweils einer eigenen Gesamtstruktur zu organisieren. Somit entstehen, ähnlich der bisher vertretenen Insellösungen, autarke Domänen, die parallel betrieben werden können und die Vorteile von Vertrauensstellungen zwischen den Domänen, die jedoch explizit eingerichtet werden müssen, und voller administrativer Hoheit in den Händen der betreibenden Struktureinheiten verbinden.

Andeutungsweise soll dies in Abbildung 6.2 aufgezeigt sein.

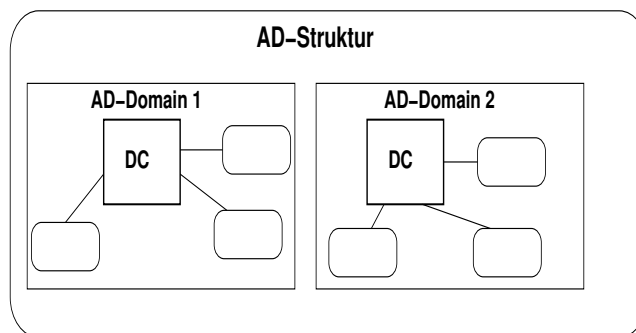


Abbildung 6.2: Parallele Active Directory Domänen

Auch hierbei wird es für sinnvoll erachtet, eine Active Directory Domain in der Verantwortung des Universitätsrechenzentrums als Dienst anzubieten, da einerseits weiterhin denkbar ist, den bisherigen LDAP-Dienst durch diesen zu ersetzen, dies weitaus interessanter jedoch im Kontext der bereitgestellten Windows-Systeme ist. Dadurch ließe sich nicht nur eine Bereitstellung von Microsoft basierender Software vorstellen oder Windows-Serverlandschaften, die speziellen Anwendungen gerecht werden, auch die Administration und Einbindung der Windows-Arbeitsplatz-Systeme kann wahrscheinlich vereinfacht werden. Außerdem lassen sich komplexe oder wiederkehrende Anpassungen an die Voraussetzungen der umgebenden Dienststruktur bei Patches, Updates oder Upgrades der Windows-Systeme vermeiden, da durch die Bereitstellung einer Active Directory Umgebung der native Einsatz von Windows-Systemen unterstützt wird. Auch die grundsätzliche Möglichkeit des Einsatzes von Gruppenrichtlinien, um beispielsweise sicherheitsrelevante Einschränkungen auf den Arbeitsplatzsystemen durchzusetzen, ist als Argument nicht zu vernachlässigen.

Active Directory Domänen in Struktureinheiten der Universität lassen sich mit voller Entfaltung der Administrator-Rechte betreiben und haben innerhalb ihres Namensraumes die Hoheit. Somit kann jede Struktureinheit in unabhängiger Entscheidung Anpassungen am eigenen AD-Schema vornehmen, Applikationen bereitstellen und Arbeitsplatzsysteme vollständig auf die eigenen Vorgaben oder Vorstellungen anpassen. Auch bei der Installation einer neuen Domäne ist kein Domänen-Administrator einer übergeordneten Domain notwendig, um die Integration zu berechtigen. Aufwendige Installationsvorgaben lassen sich vermeiden. Dabei kann aber auch der Vorteil der Zusammenarbeit über Grenzen von Struktureinheiten hinweg durch die explizite Einrichtung von Vertrauensstellungen genutzt werden. Vielleicht verbessert sich somit der Überblick der vergebenen Rechte bei Administrator-

ren von Struktureinheiten ebenfalls, da implizites Vertrauen nicht vorausgesetzt werden kann und muss. Auch ist es in diesem Falle, ebenso wie im vorherigen Beispiel, nicht notwendig, die Clientensysteme in den Namensraum der Domänen zu verschieben und beispielsweise eine Administration durch das URZ auch mittels CFEngine bleibt vorstellbar, auch wenn zusätzlich die Gruppenrichtlinien der Domäne und eventueller OUs ihre Wirkung entfalten. Nicht zuletzt bleibt die Möglichkeit der Auswahl der bereitgestellten und eingepflegten Nutzeraccounts. Dadurch behält der Domänenadmin die Kontrolle über die Auswahl der Nutzer, die sich überhaupt an seiner Domäne anmelden können.

Nachteilig bei dieser Variante des Einsatzes ist die dezentrale Verwaltung der Nutzeraccounts anzumerken, da bei dieser Betriebsvariante keinerlei Vertrauensweg via einer Root-Domain aufgebaut werden kann. Jede Windows-Domäne ist eine eigene Wurzel. Allerdings kann das beispielhaft angefügte VBScript auch diesem Umstand durch die Einschränkung der zu pflegenden Accounts durch die Liste in der *LOKNUTZER*-Datei gerecht werden. Andererseits bleibt dabei die Bereitstellung der Daten durch das URZ zu klären. Es wäre ja denkbar, ein entsprechendes File täglich aus dem Bestand der MoUSE zu generieren und im AFS bereitzustellen, jedoch bekommt somit jeder Administrator einer Active Directory Domain Zugriff² auf eine Auswahl der Daten eines jeden Nutzers im URZ. Es bliebe grundsätzlich zu klären, ob diese Datenweitergabe dem Datenschutzgesetz entspricht. Andererseits sind alle diese Daten bereits jetzt universitätsöffentlich über andere Dienste verfügbar gemacht. Trotz allem bleibt es offen, wie die Bereitstellung im Alltagsgeschäft erfolgen kann und eine Aktualisierung durch die Domänenadministratoren sichergestellt werden kann.

Ebenso ist in der Folge des eben Genannten zu beachten, dass Nutzer, die in mehreren Domänen existieren, damit faktisch mehrere Accounts haben, also auch mehrere Arbeitsplatzumgebungen pflegen müssen. Legt beispielsweise der Nutzer bei der Arbeit in der einen Domain eine Datei oder eine Verknüpfung auf seinem Desktop an, wird er diese Umgebung in einer anderen Domäne nicht so wiederfinden.

Vorteilhaft sei aber ergänzt, dass man durch diese Betriebsart den Einschränkungen, die das Heimdal-Kerberos-System derzeit gegenüber der MIT-Variante noch hat, gerecht würde.

6.2.4 Nutzung von Organisationseinheiten (OUs) für kleinere Struktureinheiten

Ein interessanter Punkt beider gerade vorgestellter Varianten ist die Möglichkeit der Bildung von Organisationseinheiten und der damit verbundenen Delegation von Berechtigungen an ansonsten normal eingeschränkte Nutzeraccounts. Entscheidet man sich grundlegend für das Angebot eines Active Directory Dienstes im Universitätsrechenzentrum, kann man vielleicht gerade durch den Aufbau einer OU-Struktur den Anforderungen und Wünschen kleinerer Struktureinheiten Genüge tun. Etwa ist es denkbar, dass ein einzelner Lehrstuhl, seine Arbeitsplatz-Systeme zwar grundsätzlich durch das URZ gepflegt haben will, aber individuelle Softwarewünsche hat, die über das Standardspektrum hinaus gehen. Dann wäre eine Lösungsvariante, die betreffenden AD-Accounts der Rechner³ in einer OU innerhalb der URZ-

²bestenfalls nur indirekt

³dort Computerkonto genannt

AD-Domäne zu vereinen und einem versierten Mitarbeiter weitergehende Berechtigungen auf dieser Auswahl von Rechnern einzuräumen oder seine Gewalt durch die Möglichkeit der Gruppenrichtlinien zu erweitern. Auch wenn dieser versierte Mitarbeiter nicht vorhanden ist, kann somit durch das URZ eine gesammelte Anpassung solcher Arbeitsgruppen-Rechner erfolgen.

Wie sich eine solche OU eingliedert, verdeutlicht Abbildung 6.3.

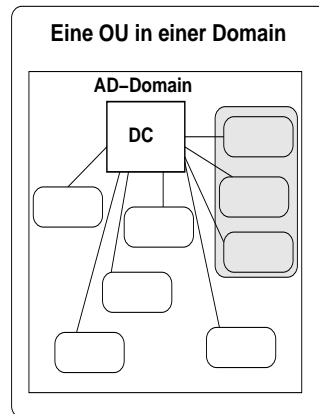


Abbildung 6.3: Organisationseinheit in einer Domäne

6.3 Lösungsvorschlag zu einer Struktur

Ursächlich durch die beiden Aspekte begründet, dass Heimdal die Zerlegung von TGS-Requests nicht unterstützt und somit ein echter Domänenbaum nur durch Umstellung auf MIT-Kerberos oder aufwendiges hausinternes Patching des Heimdal-KDCs möglich wäre, auf der anderen Seite in einem Baum alle angeschlossenen Domänen sich auf eine einheitliche Schema-Struktur einigen müssten und außerdem nur eine Exchange-Instanz in einem Baum implementierbar ist, würde ich als Autor dieser Arbeit aufbauend auf meinen bisherigen Erfahrungen empfehlen, eine derzeitige Gesamtlösung im Sinne von Einzeldomänen im parallelen Betrieb anzustreben.

Im Ergebnis sollte als Erstes eine Policy zur Einrichtung eines einheitlichen Namensraums diskutiert und erstellt werden. Dies ist meiner Ansicht nach auch unabhängig einer Entscheidung des URZs für oder gegen den Betrieb eines eigenen Active Directory sinnvoll, um Struktureinheiten der TU Chemnitz die konsequente und eine konsistente Möglichkeit einzuräumen, einen AD-Dienst in eigener Verantwortung einzurichten. Vorschläge zu einer Strukturierung dieses Namensraumes sind bereits mehrfach gefallen und sollen hier nicht noch einmal wiederholt werden. Dabei soll auch nochmal erwähnt werden, dass ausschließlich die Domaincontroller einer Active Directory Domain in diesem Namensraum eingerichtet werden müssen, da sowohl Domain-Clients als auch Mitgliedserver einer Domäne dieser angehören können, obwohl deren eigener DNS-Suffix sich von der Domäne unterscheidet.

Auf Grund der Erkenntnisse dieser Arbeit halte ich die Einrichtung einer zentralen AD-Domain für sinnvoll, um eine „saubere“ Integration von Windows-Diensten in die Struktur des URZs zu gewährleisten. In der Folge entfallen aufwändige Anpassungen an Notwendigkeiten der Kerberos-AFS-Umgebung im URZ auf den Windows Clienten Systemen aber auch beim Betrieb von Anwendungsservern. Die komplette Bereitstellung von Diensten zur Authentifizierung im Windows-Kontext kann durch die Domaincontroller der AD-Domain übernommen und sichergestellt werden. Darüberhinaus ist es dadurch möglich, die Funktion von Gruppenrichtlinien einzusetzen und somit gezieltere Eingriffe in die Betriebssystemkonfiguration von Windows-Systemen vorzunehmen, als es ausschließlich durch Registry- und Kommandozeilen-Werkzeuge möglich ist. Zusätzlich kommt neben der Möglichkeit einer Softwareverwaltung mittels CFEngine hinzu, mittels dieser Gruppenrichtlinien Software auf einer Auswahl oder allen Rechnern der Domäne zu verwalten. So wäre eine Installation oder ein Upgrade/Update der AFS-Clientensoftware durch eine zentrale Richtlinie auf allen Systemen durchzusetzen.

Als weiterer Vorteil ist zu erwähnen, dass die Einrichtung von Organisationseinheiten für Rechner einzelner Struktureinheiten vorstellbar ist, durch welche eine zentrale Administration für die grundlegenden Einstellungen der Rechner und spezielle Anpassungen an Anforderungen der Struktureinheit auch durch versierte Nutzer ermöglicht wird.

Die Domänen, welche in Struktureinheiten der Universität betrieben werden sollen, sollten als eigenständige Gesamtstrukturen angelegt werden, wodurch sich vorteilhaft nicht nur ein eigenes Schema ergibt, welches ausschließlich in der Verantwortung der Struktureinheit liegt, sondern auch eine aufwendige Integration in eine Gesamtstruktur durch Administratoren des Universitätsrechenzentrums überflüssig wird. Nachteilig ist zu erwähnen, dass sich dadurch für den Endanwender mehrere zu pflegende Umgebungen ergeben, aber auch die Absicherung der Verfügbarkeit der Domäne in der Verantwortung der Struktureinheiten liegt. Außerdem ist zu bedenken, dass sich durch die Angliederung der Nutzerverwaltung an die Datenbasis des URZs nicht nur ein einmaliger Aufwand für die Clientensysteme ergibt, sondern auch der kontinuierliche Bedarf der Aktualisierung aus diesem Datenbestand in Verantwortung des Domänenadministrators entsteht.

Beachtet werden sollte in diesem Zusammenhang vor allem die Vermittlung der Vorteile einer solchen Integration beispielsweise bei der Nutzung von anderen Diensten wie dem AFS. Immerhin gebe ich zu bedenken, dass zwar dank des Einsatzes eines zentral bereitgestellten VBScripts und der im AFS abgelegten Datendatei eine Integration auch einer eingeschränkten Nutzerauswahl relativ einfach möglich ist, aber es entspricht nicht der Erfahrung und dem täglichen Umgang eines windowsgewohnten Nutzers. Somit muss bei der Vorteilsvermittlung dem Betrieb einer eigenen Nutzerverwaltung sinnvoll und überzeugend entgegen gehalten werden.

Um eine Domäne einer Struktureinheit anzubinden, muss also seitens des Rechenzentrums nur der Namensraum geschaffen, ein KRBTGT für die Domäne im Heimdal angelegt, dessen Passwort dem Administrator der zukünftigen Domäne verfügbar gemacht und in der zentralen DNS-Konfiguration ein Forward-Eintrag oder die Dienst- und DC-Einträge erstellt werden.

Durch die Erstellung der Vertrauensstellung im Windows-System mit dem Passwort des KRBTGT aus dem Heimdal wird nicht nur eine Anmeldung der Nutzer des URZs grund-

legend ermöglicht, sondern auch Dienstanforderungen aus der Domäne an das URZ werden realisierbar, so etwa die automatische Anbindung von Homeverzeichnissen im AFS durch das „Integrated Logon“ des Windows-AFS-Clients.

Auch sollte sich über den zentralen Kerberos-Realm eine implizite Vertrauensstellung aller Nutzer mit allen Domänen ergeben. Diese kann durch die Administratoren der einzelnen Domänen mittels der Einrichtung expliziter Vertrauensstellungen untermauert werden. Dies ist bei diesem Konstrukt insofern notwendig, als das innerhalb der Domänen eine Abbildung der Prinzipale auf Nutzeraccounts der Domäne stattfinden, welche in jeder Domäne einzeln geführt werden. Über diese Vertrauensstellungen ist interstrukturelle Zusammenarbeit und der Zugriff auf Ressourcen anderer Struktureinheiten verwaltbar.

Vorteilhaft soll nochmals erwähnt werden, dass durch diesen Aufbau jede Domäne ein eigenes Schemata hat und pflegt und dieses jeweils nur auf die eigenen Bedürfnisse anpassen muss. Somit kann innerhalb der Struktureinheiten frei über die Auswahl der anzubietenden Dienste über diesen Server entschieden und auch die Softwareauswahl frei gestaltet werden. Es bedarf keiner Abstimmung bei Erweiterungen mit anderen Domänen der Gesamtstruktur und es entstehen auch anderen Domänenadmins keine Aufwände durch Wünsche fremder Domänen. Nicht zuletzt wird es ermöglicht, mehrere Instanzen der MS-Collaborations-Software „Exchange“ entsprechend der Wünsche und Vorstellungen der einzelnen Struktureinheiten zu betreiben.

Ein hier letztgenannter Vorteil ist der Aufbau eines globalen Katalogs für jede Gesamtstruktur, also jede Domain. Dadurch ergibt sich ein überschaubares Wachstum des globalen Katalogs und die Chance auch in der zentralen Domäne, wo mindestens ein globaler Katalog zu führen ist, den Datenbestand, welcher jeweils in den Katalog repliziert wird, zu überwachen. In einer gesamten Baumstruktur hingegen ist es schwerlich zu überblicken, wieviele Daten aus den Verzeichnissen der angegliederten Domänen zu erwarten sind.

Grundsätzlich vorstellbar ist auch eine Baumstruktur, wobei zu jeder Domain ein Heimdal-Cross-Realm-Trust bestehen muss und trotzdem in jeder Domain die Nutzer zu pflegen sind. Dadurch entstehen implizite Vertrauensstellungen innerhalb eines kleinen einheitlichen Namensraumes. Nachteilig entsteht neben dem administrativen Aufwands bei der Integration der neuen Subdomänen auch wieder die Einschränkung auf ein einheitliches Gesamtstrukturschema.

6.4 Aufbau einer zentralen Domäne

Hier folgen die Schritte, welche notwendig erscheinen, um eine zentrale Active Directory Domain aufzusetzen. Grundlegend unterscheidet sich der Weg nur geringfügig von den Notwendigkeiten, um eine Subdomäne aufzusetzen. Nach der Auswahl adäquater Serverhardware mit wie erwähnt mindestens redundanter Netzwerkanbindung und ausreichender Festplattenkapazität für die Verzeichnisdatenbank und den globalen Katalog, wird auf dieser ein Windows Server 2003 installiert. Je nachdem, welchem DNS-Modell gefolgt wird, sollten die DNS-Einträge für die Domäne, mindestens den Domaincontroller (oder eben für die DCs) und dessen Dienste, entsprechend der Anlage A angelegt beziehungsweise der Forward-Eintrag für die Domäne auf den DC eingerichtet werden. Beim Eintrag entsprechend der Anlage ist zu beachten, dass alle Einträge für jeden Domaincontroller einer Do-

mäne erzeugt werden müssen, einzig der `_ldap._tcp.pdc._msdcs` nur auf den entsprechenden Betriebsmaster zeigen darf (üblicherweise der erste DC einer Domäne, wenn nicht manuell verschoben) und die Einträge mit „gc“ nur für GC-Server erzeugt werden dürfen. Außerdem kann bereits ein entsprechendes Prinzipal „krbtgt/windows.domain@TU-CHEMNITZ.DE“ im Heimdal mit manuellem Passwort eingerichtet werden.

Nach der grundlegenden Installation wird der Server zum Domaincontroller für den zentralen Windowsnamensraum, beispielsweise „ad.tu-chemnitz.de“ gehoben. Dabei wird der Server, entsprechend Abbildung 6.4, als neuer Domaincontroller einer Gesamtstruktur deklariert und der Domäne der Name zugeordnet.

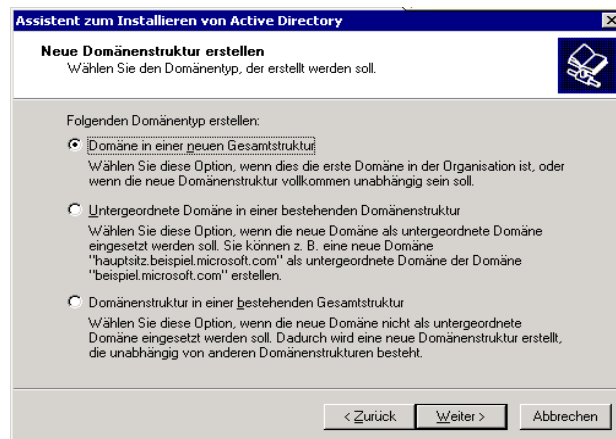


Abbildung 6.4: Einrichtung einer neuen Gesamtstruktur

Es folgt ein Test, der Funktionalität des DNS-Servers. Wird sich für das Modell des statischen DNS mittels BIND entschieden, kann der Hinweis zu den abgewiesenen dynamischen Updates ignoriert werden, wird statt dessen mittels DNS-Forward in der zentralen Konfiguration gearbeitet, empfiehlt sich die grundlegende Installation eines neuen Domaincontrollers inklusive des Microsoft DNS-Servers. Nach Anlegen des Active Directorys und dem Neustart des Domaincontrollers ist die Domäne verfügbar. Bei diesem Neustart werden gegebenenfalls auch die Einträge im dynamischen DNS für die Dienste des Servers erzeugt. In der Folge können die entsprechenden Nutzer angelegt werden, entweder per Skript viele automatisiert oder manuell über das MMC-SnapIn „Active Directory Benutzer und Computer“, wobei zu beachten ist, dass die Zuordnung zum entsprechenden Kerberos-Prinzipal einzurichten ist. Dies ist erst nach Aktivierung der erweiterten Funktionen im Menüpunkt „Ansicht“ durch die Aktion „Namenszuordnungen“ im Tab „Kerberos-Namen“ möglich. Das VBScript, welches in grundlegenden Zügen im Anhang zu finden ist, richtet Nutzeraccounts mit der entsprechenden Eigenschaftszuordnung im Active Directory ein. Um dieses Skript sinnvoll nutzen zu können, ist eine Bereitstellung der *NUTZER*-Datei für den Domaincontroller lesbar notwendig. Denkbar ist hierbei eine Bereitstellung im AFS, wobei dies wahrscheinlich nur mittels IP-basierter Rechtevergabe sinnvoll möglich ist. Welche Mechanismen angewandt werden, um eine Aufbereitung der Daten aus der MoUSE und deren Bereitstellung im AFS zu realisieren, wird hier nicht weiter betrachtet, da vergleichbare Implementierungen bereits im URZ vorhanden und im Detail von der Entscheidung abhängig

sind, welche Daten zur Bereitstellung ausgewählt werden. Grundsätzlich ist denkbar, diese Bereitstellung mittels eines Servers der MoUSE-Umgebung mittels Cronjob umzusetzen. Auch auf den Domaincontrollern ist mittels geplantem Task die Ausführung des VBScripts möglich.

Außerdem sollte die Funktionsebene der Gesamtstruktur beziehungsweise der Domäne an dieser Stelle festgelegt werden. Da alle Domänen in diesem Konzept unabhängig sind, aber durch entsprechendes Heraufstufen der Funktionsebene verbesserte Features der Systeme nutzbar werden, sollte mindestens die Funktionsebene „Windows 2000 pur“ gewählt werden. Damit sind universelle Gruppen verwendbar, aber auch noch Windows 2000 Domaincontroller einsetzbar. Ist auch dies nicht geplant, kann beispielsweise durch noch weiteres Heraufstufen der Replikationsaufwand seitens des globalen Katalogs vermindert werden, da ja unter Windows 2003 nur noch Änderungen propagiert werden.

Um diese Domäne als Dienst zu realisieren, muss weiterhin deren möglichst hohe Verfügbarkeit gesichert werden, um die Authentifizierung der angeschlossenen Clienten-Systeme, aber auch die sonstig durch die Domäne angebotenen Dienste zu gewährleisten. Für die Authentifizierung ist die Verfügbarkeit eines GC-Servers und eines Domaincontrollers der Domäne notwendig, an der sich ein Nutzer anmelden will. Daher ist es empfehlenswert, mehr als einen GC-Server und sowieso mehr als einen Domaincontroller zu betreiben. Der Grund, weshalb standardmäßig nur ein globaler Katalog innerhalb einer AD-Domain eingerichtet wird, kann in dem entstehenden Replikationsaufwand zwischen GC-Servern begründet liegen. Wenn allerdings durch die Sicherstellung einer breitbandigen Anbindung, wenn nicht gar durch die direkte Verbindung zweier Domaincontroller via der Redundanzverkabelung, die Menge des Netzwerkverkehrs zwischen diesen beiden wenig bis keine Rolle spielt, ist es mehr als sinnvoll, mehreren DCs die Funktion des globalen Katalogs zuzusprechen. Dazu muss im MMC-SnapIn „Active Directory-Standorte und Dienste“ am Standort des Servers, in den NTDS-Settings des entsprechenden Servers der Haken im globalen Katalog gesetzt werden, wie in Abbildung 6.5 zu sehen.

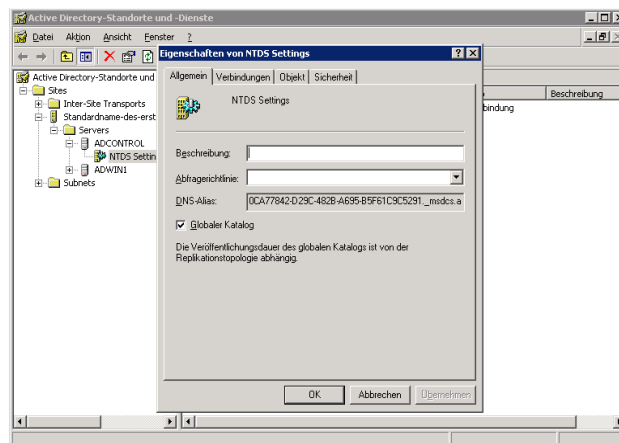


Abbildung 6.5: Einstellungen zum globalen Katalog

6.5 Domänen bei Struktureinheiten

Hier sollen nur kurze Kommentare zu einer möglichen Integration einer Domäne in der Verantwortung einer Struktureinheit genannt werden. Einerseits werden die Punkte aufzählt, die bei der Erstellung eines Gesamtkonzepts zum einheitlichen Betrieb von Active Directory Domänen im Campusnetz der TU Chemnitz mit besonderem Augenmerk bedacht werden sollen, andererseits soll eine Vorlage für die Entwicklung einer Handreichung für neue Domänenadministratoren im Campusnetz entwickelt werden.

Auf Grund der bisherigen steten Abwägung beider oben genannter Alternativen und um trotz des eingebrachten Gesamtvorschlags einer Entscheidung seitens des URZ für die eine oder andere Variante nicht vorzugreifen, werden auch hier noch beide Varianten betrachtet und gegenübergestellt.

Auf die Einstellungen, die sich in Folge der Entscheidung für die eine oder andere Variante ergeben, soll später eingegangen werden.

Grundsätzlich können die Hinweise zur Installation kurz gehalten werden, da die Basisaktionen in der reichlichen Fülle der Literatur zum Thema „Active Directory“ und „Windows Server“ ausgiebig und umfassend erläutert werden.

Hinsichtlich der Einstellungen zum DNS-System kann pauschal gesagt werden, dass eine Konfiguration für den Zugriff auf die zentralen DNS-Server für alle Aspekte ausreichend ist. Sollte sich für die Einrichtung eines DNS-Forwards für eine Domäne entschieden werden, kann jeder Rechner des Campusnetzes eben durch diesen Forward auch die Domäne und ihre Dienste erreichen. Wird hingegen die statische Konfiguration im zentralen BIND-System präferiert, muss sogar durch Zugriff aller Systeme auf das zentrale DNS-System die Lokalisierung der Dienste sichergestellt werden. Bei der erstgenannten Variante des DNS-Einsatzes ist zu beachten, dass bei der Installation des Domaincontrollers am sinnvollsten dem Wizard zur Standardinstallation eines DCs gefolgt werden sollte, um auch den DNS-Server mit zu installieren. Hat man dies allerdings vergessen oder stellt erst später um, werden die Einträge im DNS ebenfalls angelegt, sobald der DC neu gestartet wird. Um den Microsoft DNS-Server gegebenenfalls nachträglich zu installieren, muss dieses über den Punkt „Software“ der Systemsteuerung erfolgen. Ist der DNS-Server installiert, muss als einzige Konfiguration die Weiterleitung aller durch ihn nicht realisierbaren Anfragen an die zentralen DNS-Server des URZs eingerichtet werden. Dazu wird die DNS-Verwaltungskonsolle geöffnet und in den Eigenschaften des DNS-Servers diese Weiterleitung eingetragen, wie es in Abbildung 6.6 angedeutet ist.

Ein weiterer wesentlicher Punkt, der bei der Konzeptionierung von zu integrierenden Subdomänen Beachtung finden sollte, ist die Findung und Dokumentation einer Handlungs- und Arbeitsanweisung, um eine eigene Domäne zu integrieren. Zu Aspekten der AD-Integration folgen noch Aussagen, da diese von Variante zu Variante variieren, aber gemein haben beide Varianten die Notwendigkeit, der Zuteilung eines DNS-Namens und die Einrichtung der entsprechenden BIND-Konfigurationen für diese neue Domain sowie die Einrichtung des Kerberos-Realm-Trust-Prinzipals in der Heimdal-Datenbank. Diese Schritte sollten durch den Administrator der neuen Subdomäne beim URZ ausgelöst und die Ergebnisse durch dieses in adäquater Form bereitgestellt werden, um ihm die Installation seiner Domäne zu ermöglichen.

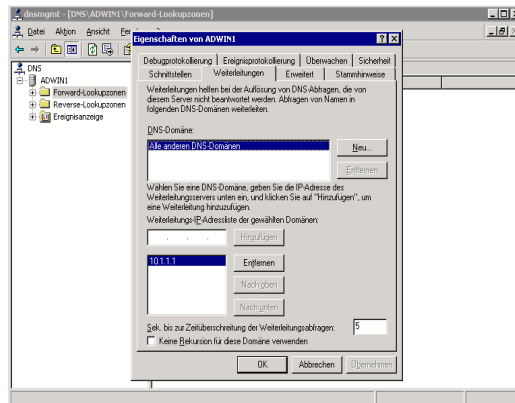


Abbildung 6.6: Einrichtung der DNS-Weiterleitungen im MS-DNS-Server

Bei der Integration in einen Domänenbaum (also der ersten Variante) gibt es noch einen wesentlichen Punkt im Workflow, der durch eine zentrale Regelung abgedeckt werden sollte.

6.5.1 Integration in einen Domänenbaum

Beim Betrieb eines Domänenbaums, in den eine weitere Domäne integriert werden soll, muss im Rahmen des Integrationsprozesses bedacht werden, dass einer der Schritte der Eintrag der Subdomäne in das Active Directory der hierarchisch übergeordneten Domäne ist, was Rechte des Administrators dieser Domäne erfordert.

Standardmäßig gibt es mehrere Administrator-Rollen in einer Domäne, etwa der Konten-Operator oder der Organisations-Admin. Im Rahmen der Tests wurde versucht, mit einer solchen gegenüber dem regulären Administrator eingeschränkten Rolle eine Integration einer Subdomäne vorzunehmen, was leider misslang. Somit ist im Rahmen der Propagierung eines Windows-Server zum ersten Domaincontroller einer neuen Subdomäne der Eingriff beziehungsweise die Berechtigung des Administrators der darüber angesiedelten Domäne nötig. Prinzipiell wäre es möglich, dem Administrator der Subdomäne durch Bereitstellung eines entsprechend berechtigten Nutzeraccounts die Fähigkeit dazu zu geben, doch wird es in den wenigsten Anwendungsfällen gewünscht und sinnvoll sein.

Die Lösungen, die sich im Falle dieses Problem es anbieten, sind einerseits die grundlegende Installation und Integration durch einen Administrator der Stammdomäne und erst danach folgenden Auslieferung und Übergabe an den Administrator der Subdomäne beziehungsweise andererseits wäre denkbar, die Grundinstallation durch den zukünftigen Subdomänen-Administrator durchführen zu lassen, ihn mit der Einrichtung eines qualifizierten Admin-Kontos für den Administrator der Stammdomäne und den Remotezugriff beispielsweise via RDP einzurichten, so dass dieser Stamm-Administrator in der Folge der Grundinstallation die Propagierung auch aus der Ferne vornehmen kann. Danach ist die Löschung des Stamm-Admin-Kontos durch den Subdomänen-Administrator wieder möglich, um die administrative Hoheit zu wahren. Gleiches Vorgehen wäre beim Löschen einer Domäne zu beachten. Die Integration einer neuen Substruktur zeigt Abbildung 6.7.

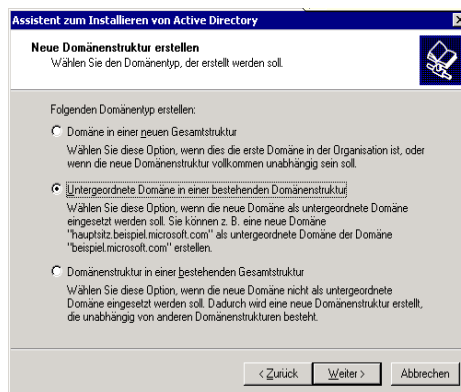


Abbildung 6.7: Einbindung einer neuen Subdomäne

Die Integration eines neuen Domaincontrollers in eine bestehende Domäne hingegen ist unproblematisch, da dazu nur Eintragungen in das Active Directory der eigenen Domäne erfolgen, dies also durch jeden Domänen-Administrator der eigenen Domäne erfolgen kann. Gleicher Aufwand entsteht bei der Einrichtung paralleler Domänen einer bestehenden Gesamtstruktur, was beispielsweise für Institutionen interessant ist, die mehrere Domänen unter einer administrativen Hoheit betreiben.

Als „Nebeneffekt“ der Integration wird die implizite Vertrauensstellung sowohl in der Stamm- als auch Subdomäne eingetragen, was keinen weiteren Eingriff erfordert, um Benutzer der selben Gesamtstruktur für die Nutzung von Ressourcen der eigenen Domäne zu berechtigen. Einzig die Rechtevergabe auf der Ressource muss erfolgen, gegebenenfalls ist die Strukturierung der Nutzer oder Ressourcen in Gruppen notwendig.

6.5.2 Erstellen eigener Domänen

Nach der grundlegenden Installation des Serverbetriebssystems wird dieser Server zum Domaincontroller propagiert. Dabei wird jeweils eine neue Domäne in einer neuen Gesamtstruktur erstellt. Hierbei gelten die Hinweise, welche bereits bei der zentralen Domäne gegeben wurden. Es ist also zu beachten, ob ein DNS-Server mit zu installieren ist und wie der zugewiesene Namensraum, also der Name der Domäne lautet. Ist die Domäne grundsätzlich eingerichtet, wird unter „Active Directory Standorte und Vertrauensstellungen“ die explizite Vertrauensstellung mit dem Kerberos-Realm der Universität eingerichtet. Diese sollte ausgehend sein, also Nutzern des Kerberos-Realms wird vertraut, und mit dem durch das URZ übermittelte Passwort eingerichtet sein. Eingehende Vertrauensstellungen würden seitens des Kerberos ignoriert, da das entsprechende KRBTGT in der Heimdal-Datenbank fehlt, um Tickets der Windows-Domäne zu validieren. Dieses lautet „krbtgt/TU-CHEMNITZ.DE@windows.domain“. Dadurch wird verhindert, dass Nutzer, die nur in einer Windows-Domäne existieren, sich an URZ-Maschinen authentifizieren können.

Ist die Zusammenarbeit mit Nutzern anderer Windows-Domänen gewünscht, muss auch zu diesen eine entsprechende Vertrauensstellung in Rücksprache mit dem dortigen Administrator eingerichtet werden, um deren Nutzer in der eigenen Domain zu berechtigen oder eigene

Nutzer in der fremden Domain zu befähigen. Auch dies erfolgt im oben genannten MMC-SnapIn.

Der zweite wesentliche Punkt der Einrichtung einer solchen Subdomäne ist die Verwaltung der Nutzeraccounts. Diese muss in der Domäne selbst erfolgen, da im Rahmen der Authentifizierung der Heimdal-KDC direkt an den Domaincontroller der Windowsdomäne vermittelt, nachdem er den Nutzer authentifiziert hat. Dort wird dann nach einem Nutzer gesucht, der in seinen Eigenschaften eine Namenszuordnung zum authentifizierten Prinzipal hat. Diese Namenszuordnung muss, wie bereits mehrfach erwähnt, bei der Einrichtung des Nutzers angelegt werden. Wie in der Arbeit bereits angeführt, gibt es nur zwei Möglichkeiten, adäquate Nutzer in der Domain anzulegen. Die Eine liegt in der Einrichtung via des GUIs und der Namenszuordnung in den erweiterten Funktionen. Die Andere besteht aus der Nutzung eines entsprechenden VBScripts, welches mittels des Windows Scripting Hosts die Nutzer im Active Directory anlegt und die entsprechenden Verknüpfungen in den Objekten einfügt. Beispielhaft wurde ein funktionierendes Skript als Anlage angehängt. Um allen Anforderungen des täglichen Betriebs gerecht zu werden, ist sicher noch die eine oder andere Anpassung notwendig, aber eine grundsätzliche Funktionalität ist gewährleistet. Dabei sollte die Ausführung des Verfahrens automatisiert eingerichtet werden, um die Aktualität der Daten in den Domänen zu gewähren, aber auch um den Administrator einer Subdomäne zu entlasten. Denkbar ist die Einrichtung entsprechender „geplanter Tasks“, die das entsprechende VBScript auf dem Domaincontroller mit den Rechten eines Domänenadministrators ausführen.

6.6 Einstellungen an den Clienten

Als Abschluss sollen noch ein paar Hinweise zur Konfiguration der Clientensysteme, egal ob Windows XP, Windows 2000 oder Windows 2003, untergebracht werden. Diese stehen im Zusammenhang mit den notwendigen Einstellungen, um sie in einer Umgebung zu betreiben, in der gegen einen externen Kerberos-Server authentifiziert werden soll. Es geht aber auch nochmal um die Konfiguration des DNS-Zugriffs auf den Clienten.

Wie bereits mehrfach erwähnt, ist es nicht notwendig, dass ein Domänenmitglied im selben Namensraum steht, wie die Domäne. So ist es möglich, einen Clienten „host.informatik.tu-chemnitz.de“ in eine Domäne „ad.hrz.tu-chemnitz.de“ zu integrieren. Dazu muss nur die Kopplung zwischen der angeschlossenen Domäne und dem DNS-Suffix aufgehoben werden, was mittels des Löschens eines Hakens in den erweiterten Einstellungen des Computernamen-Tabulators in den Systemeigenschaften der Systemsteuerung möglich ist, wie Abbildung 6.8 zeigt.

Weiterhin sollte jedes Clientensystem im Campusnetz so konfiguriert werden, dass es auf die zentralen DNS-Serverkomponenten zugreift. Selbst bei der Entscheidung für den Betrieb von dezentralen DNS-Servern für die einzelnen Subdomänen, wird das zentrale DNS-System die sinnvolle Zuweisung zu den entsprechenden Servern sicherstellen. Sind Clienten aber für den Zugriff auf beispielsweise einen Subdomänen-DNS konfiguriert, verursacht dessen eventuelle Fehlkonfiguration erhebliche Probleme für alle angeschlossenen Systeme. Daher

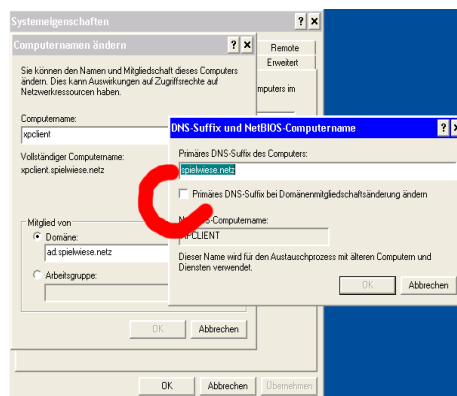


Abbildung 6.8: Entkopplung von DNS-Suffix und Domainnamen

sei hier nochmal auf die Einrichtung der DNS-Weiterleitung verwiesen.

Ein letzter aber wesentlicher Punkt der Konfiguration der Clientensysteme ist die Einrichtung der Authentifizierung gegen einen externen Kerberos-Server. Dazu stehen, wie in 5.6.2 erwähnt, zwei denkbare Wege zur Verfügung:

- Einrichtung mittels *ksetup*
- Eintragung in der Registry

Ksetup ist Bestandteil der Support Tools von Windows 2003 und Windows XP und wird nur bei dessen vollständigen Installation bereitgestellt. Die Syntax ist einfach und im Groben reicht der Aufruf

```
ksetup /AddKdc KERBEROS-REALM kdc.domain.tld
```

für jeden KDC des externen Kerberos-Realm, um die Einrichtung vorzunehmen. So wäre es beispielhaft

```
ksetup /AddKdc TU-CHEMNITZ.DE kerberos.tu-chemnitz.de
ksetup /AddKdc TU-CHEMNITZ.DE kerberos-1.tu-chemnitz.de
ksetup /AddKdc TU-CHEMNITZ.DE kerberos-2.tu-chemnitz.de
```

für den Heimdal-Kerberos-Realm des URZ. Damit wird im Anmeldefenster des entsprechenden Rechners eine Anmeldedomain hinzugefügt, die den Namen des Kerberos-Realms trägt und als solche ausgewiesen ist und ihre Authentifizierungsanfragen an einen verfügbaren KDC der Liste sendet.

Die Alternative besteht aus der Einrichtung der notwendigen Einträge für diese Anmeldedomain direkt in der Registry. Auch hierfür sind mehrere Wege vorstellbar. So können die Einträge etwa manuell mittels des *regedit*-Werkzeugs angelegt werden oder diese Arbeit einmalig erfolgen, dann die Änderungen exportiert und mittels *.reg*-File bereitgestellt werden, wodurch sie auf weiteren Clienten einfach durch Aufruf des Files erzeugt werden

können. Dieses reg-File könnte seitens des URZ zentral bereitgestellt werden. Als weitere Form der Einrichtung wäre eine Bereitstellung eines msi-Paketes mit dem angesprochenen reg-File zentral durch das URZ vorstellbar. Somit könnte die Konfiguration aller Clienten einer Domäne sogar durch eine Gruppenrichtlinie erfolgen, in der dieses msi-Paket den Rechnern zugewiesen würde. Sogar Anpassungen an eventuell auftretende Veränderungen der Kerberos-Server-Landschaft wäre mittels eines neuen Pakets und dessen Zuweisung mittels GPO einfach realisierbar.

Unabhängig von der Methode ist es notwendig die folgenden Registry-Einträge zu erzeugen, um eine neue Anmeldedomain hinzuzufügen:

Erzeugung eines neuen Schlüssels:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\_
Domains\KERBEROS – REALM
```

und innerhalb dieses des folgenden Wertepaares:

```
KdcNames REG_MULTI_SZ kdc.domain.tld
```

Also erneut beispielhaft am TU-CHEMNITZ.DE-Realm des URZ wäre dies

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\_
Domains\TU – CHEMNITZ.DE
```

und darin:

```
KdcNames REG_MULTI_SZ kerberos.tu-chemnitz.de \
kerberos-1.tu-chemnitz.de kerberos-2.tu-chemnitz.de
```

6.7 Standorte

Abschließend sei noch kurz auf die Strukturierung mittels Standorten eingegangen. Standorte bieten, wie eingangs erläutert, eine gute Möglichkeit der weiteren Strukturierung der Domänen. Dabei sollen vor allem geographische Aspekte mit bedacht werden, die sich in der Vernetzungstopologie widerspiegeln. Es sei allerdings hier darauf verwiesen, dass viele Vorteile der Standorte bei Nutzung statischen DNS' verloren gehen, wenn dieses nicht auf die Standorte eingeht. Sollte beispielsweise eine Strukturierung in die Campusteile „Straße der Nationen“ und „Reichenhainer Straße“ angedacht werden, muss auch dafür gesorgt werden, dass die Clienten der Reichenhainer Straße auf die Domaincontroller und vor allem die GC-Server dort vor Ort zugreifen. Gleiches gilt für LDAP-Clienten. Dazu muss die DNS-Konfiguration um die im Theorieteil erwähnten *Standort*-Komponenten erweitert werden und die Zuweisung der Computerkonten zum entsprechenden Standort erfolgen. Es bleibt von den Erfahrungen erster Einsätze abhängig, in wie weit sich hier eine Strukturierung in Standorte lohnenswert abzeichnet.

Bei der Nutzung von Standorten wird zwischen den Standorten eine Replikation der Domaindaten mittels TCP/IP oder SMTP angeboten, was in den „Active Directory Standorten“ unter den NTDS-Settings des Standortes einzurichten ist und die Gestaltungsmöglichkeit einer DMZ erhöhen soll. Auch werden dann verschiedene Einstellungen zu Update-Intervallen einstellbar. Da dies nochmals ein komplexes Thema ist, dessen Nutzen hier vorerst nicht abzuschätzen ist, wird erneut auf entsprechende Literatur wie [KT04] oder [Tie03] verwiesen.

Literaturverzeichnis

- [Act00] *Active Directory Architecture*, Sept. 2000,
<http://www.microsoft.com/technet/prodtechnol/windows2000serv/technologies/activedirectory/deploy/projplan/adarch.msp>.
- [Act01] *User Attributes - Inside Active Directory*, Dez. 2001,
<http://www.kouti.com/tables/userattributes.htm>.
- [Act05] *Scripts and Files - Inside Active Directory*, Sept. 2005,
<http://www.kouti.com/scripts.htm>.
- [ADT06] *IADsLargeInteger Interface*, Jan. 2006,
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ads/ads/iadslargeinteger.asp>.
- [Alb05] *Active Directory Service at University of Alberta*, Okt. 2005,
<http://www.ualberta.ca/AICT/auth/ADS.htm>.
- [All03] Robbie Allen: *Active Directory Cookbook*, O'Reilly Verlag, Sebastopol, CA, USA, 2003, ISBN 0-596-00464-8.
- [Ano04] *How to configure Active Directory to allow anonymous queries*, Nov. 2004,
<http://support.microsoft.com/default.aspx?scid=kb;en-us;Q320528>.
- [Ano05] *Anonymous LDAP operations to Active Directory are disabled on Windows Server 2003 domain controllers*, Febr. 2005,
<http://support.microsoft.com/default.aspx?scid=kb;en-us;326690>.
- [BIN05] *ISC BIND*, Sept. 2005,
<http://www.isc.org/sw/bind/>.
- [Bod05] Ulrich B. Boddenberg: *Konzepte und Lösungen für Microsoft-Netzwerke*, Galileo Press Verlag, Bonn, 2005, ISBN 3-89842-663-7.
- [Car03] Gerald Carter: *LDAP System Administration*, O'Reilly Verlag, Sebastopol, CA, USA, 2003, ISBN 1-56592-491-6.
- [Col00] *Kerberos Tests at University of Colorado*, Mai 2000,
<http://www.colorado.edu/its/windows2000/>.
- [Coo03] *Active Directory Cookbook Source Code*, Sept. 2003,
<http://www.rallenthome.com/books/adcookbook/code.html>.

- [DCE05] *DCE Project*, Sept. 2005,
<http://www.opengroup.org/dce/>.
- [DDN97] *RFC DDNS*, Apr. 1997,
<http://www.faqs.org/rfcs/rfc2136.html>.
- [DNS96] *RFC DNS SRV*, Okt. 1996,
<http://www.faqs.org/rfcs/rfc2052.html>.
- [faq05] *Anlegen zahlreicher Benutzer zu Testzwecken*, Mai 2005,
[http://www.faq-o-matic.net/content/view/65/48/
#vieluser](http://www.faq-o-matic.net/content/view/65/48/#vieluser).
- [Gar03] Jason Garman: *Kerberos - The Definitive Guide*, O'Reilly Verlag, Sebastopol, CA, USA, 2003, ISBN 0-596-00403-6.
- [Glo05] *Globus Project*, Sept. 2005,
<http://www.globus.org/>.
- [GPM05] *Gruppenrichtlinien-Verwaltungskonsole mit Service Pack 1*, Dez. 2005,
[http://www.microsoft.com/downloads/
details.aspx?displaylang=de\&FamilyID=
0A6D4C24-8CBD-4B35-9272-DD3CBFC81887](http://www.microsoft.com/downloads/details.aspx?displaylang=de\&FamilyID=0A6D4C24-8CBD-4B35-9272-DD3CBFC81887).
- [Gru05] *Microsoft TechNet AD Gruppenbereich*, Jan. 2005,
[http://www.microsoft.com/technet/prodtechnol/
windowsserver2003/de/library/ServerHelp/
79d93e46-ecab-4165-8001-7adc3c9f804e.mspx](http://www.microsoft.com/technet/prodtechnol/windowsserver2003/de/library/ServerHelp/79d93e46-ecab-4165-8001-7adc3c9f804e.mspx).
- [Hei05] *Webseite des Heimdal Projekts*, Nov. 2005,
<http://www.pdc.kth.se/heimdal/>.
- [it-03] *Benutzer per Script im Active Directory anlegen*, Dez. 2003,
[http://www.it-academy.cc/content/article_browse.php?
ID=1173](http://www.it-academy.cc/content/article_browse.php?ID=1173).
- [Ker04] *Kerberos Protokoll Registrierungseinträge und KDC-Konfigurationsschlüssel in Windows Server 2003*, Okt. 2004,
[http://support.microsoft.com/default.aspx?scid=kb;de;
837361](http://support.microsoft.com/default.aspx?scid=kb;de;837361).
- [Ker05] *Kerberos Declarations*, Nov. 2005,
<http://www.kerberos.isi.edu/>.
- [kse03] *Microsoft TechNet KSetup Overview*, März 2003,
[http://www.microsoft.com/technet/prodtechnol/
windowsserver2003/library/TechRef/
225569a0-dc77-452e-bc36-bdfd40001a90.mspx](http://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/TechRef/225569a0-dc77-452e-bc36-bdfd40001a90.mspx).
- [KT04] Stephanie Knecht-Thurmann: *Active Directory*, Addison-Wesley Verlag, München, 2004, ISBN 3-8273-2130-1.

- [Mic03] Microsoft: *Entwerfen und Einführen von Active Directory- und Sicherheitsdiensten für Windows Server 2003*, MS Press Verlag, Unterschleißheim, 2003, ISBN 3-86063-419-4.
- [Mic05] *Windows Services at University of Michigan*, Okt. 2005,
<http://www.umich.edu/~lannos/windows/>.
- [MIT05] *Kerberos: The Network Authentication Protocol*, Dez. 2005,
<http://web.mit.edu/kerberos/www/>.
- [MoU01] *Mitteilungen des URZ 1/2001*, Febr. 2001,
<http://archiv.tu-chemnitz.de/pub/2001/0014/data/index.html>.
- [MoU05] *MoUSE User Interface*, Dez. 2005,
<https://mouse.hrz.tu-chemnitz.de/user/>.
- [MR99] Klaus Schmidt Matthias Reinwarth: *Verzeichnisdienste*, VDE Verlag, Berlin, Offenbach, 1999, ISBN 3-8007-2373-5.
- [Ope05a] *OpenAFS Project*, Sept. 2005,
<http://www.openafs.org/>.
- [Ope05b] *OpenLDAP Project*, Sept. 2005,
<http://www.openldap.org/>.
- [Que05] *Quest Recovery Manager for Active Directory*, Okt. 2005,
<http://wm.quest.com/products/recoverymanagerad/>.
- [RA04] Alistair G. Lowe-Norris Robbie Allen: *Active Directory*, O'Reilly Verlag, München, 2004, ISBN 3-89721-173-4.
- [Sam05] *Webseite des SAMBA Projekts*, Dez. 2005,
<http://www.samba.org/>.
- [Sch03] Ulrich Schlüter: *Integrationshandbuch Microsoft-Netzwerke*, Galileo Press Verlag, Bonn, 2003, ISBN 3-89842-402-2.
- [Ses05] *SESAME Project*, Sept. 2005,
<http://www.cosic.esat.kuleuven.ac.be/sesame/>.
- [Sta03] William Stallings: *Betriebssysteme - Prinzipien und Umsetzung*, Pearson Studium Verlag, München, 2003, ISBN 3-8273-7030-2.
- [Thu05] *Mozilla - Thunderbird*, Dez. 2005,
<http://www.mozilla.com/thunderbird/>.
- [Tie03] Eric Tierling: *Windows Server 2003*, Addison-Wesley Verlag, München, 2003, ISBN 3-8273-2076-3.
- [Tul04] Mitch Tulloch: *Windows Server Hacks*, O'Reilly Verlag, Sebastopol, CA, USA, 2004, ISBN 0-596-00647-0.

- [Was01] *Implementation of Crossrealm Referral Handling in the MIT Kerberos Client*, Febr. 2001,
<http://www.cs.washington.edu/homes/mikesw/papers/xrealm.pdf>.
- [Was05] *Windows Domains at the University of Washington*, Okt. 2005,
<http://www.washington.edu/computing/support/windows/UWdomains/index.html>.
- [Wel05] Tobias Weltner: *Scripting für Administratoren*, MS Press Verlag, Unterschleißheim, 2005, ISBN 3-86063-979-X.
- [Win00a] *Windows Gruppen und Berechtigungen*, Sept. 2000,
<http://www.msxfaq.de/verschiedenes/gruppen.htm>.
- [Win00b] *Step-by-Step Guide to Kerberos 5 (krb5 1.0) Interoperability*, Jan. 2000,
<http://www.microsoft.com/technet/prodtechnol/windows2000serv/howto/kerbstep.mspx>.
- [Win05] *Windows Server 2003 Service Pack 1 (SP 1)*, März 2005,
<http://www.microsoft.com/germany/windowsserver2003/technologien/sp1/default.mspx>.
- [XAD05] *XAD von PADL Software Pty Ltd*, Sept. 2005,
<http://www.padl.com/Products/XAD.html>.

Selbstständigkeitserklärung

Hiermit erkläre ich, daß ich die vorliegende Arbeit selbstständig angefertigt, nicht anderweitig zu Prüfungszwecken vorgelegt und keine anderen als die angegebenen Hilfsmittel verwendet habe. Sämtliche wissentlich verwendete Textausschnitte, Zitate oder Inhalte anderer Verfasser wurden ausdrücklich als solche gekennzeichnet.

Chemnitz, den 25. März 2006

Damaschke Marko

Anhang A

Bindkonfiguration der Teststellung

A.1 named.conf

```
zone "sub.ad.spielwiese.netz" IN {
    type forward;
    forward first;
    forwarders { 10.1.1.20; };
};
zone "ad.spielwiese.netz" IN {
    type master;
    file "/etc/bind/db.ad.spielwiese.netz";
};

zone "spielwiese.netz" IN {
    type master;
    file "/etc/bind/db.spielwiese.netz";
};

zone "1.1.10.IN-ADDR.ARPA" IN {
    type master;
    file "/etc/bind/db.10.1.1";
};
```

A.2 db.spielwiese.netz

```
localhost                IN      A          127.0.0.1
_kerberos.spielwiese.netz. IN      TXT       "SPIELWIESE.NETZ"
_kerberos._udp.SPIELWIESE.NETZ. IN      SRV      1 0 88 ying.spielwiese.netz.
_kerberos._tcp.SPIELWIESE.NETZ. IN      SRV      1 0 88 ying.spielwiese.netz.
_kerberos-adm._tcp.SPIELWIESE.NETZ IN      SRV      1 0 749 ying.spielwiese.netz.
_kpasswd._udp.SPIELWIESE.NETZ. IN      SRV      1 0 464 ying.spielwiese.netz.
_kerberos._tcp.dc._msdcs.spielwiese.netz. IN      SRV      1 0 88 ying.spielwiese.netz.
ying                     IN      A          10.1.1.1
ying                     IN      TXT       "Firewall, Gate & Kerberos"
datenbank                IN      A          10.1.1.2
datenbank                IN      TXT       "MySQL & LDAP"
```

A.3 db.ad.spielwiese.netz

```
localhost                IN      A          127.0.0.1
_kerberos._tcp.dc._msdcs.ad.spielwiese.netz. IN      SRV      1 0 88 adcontrol.ad.spielwiese.netz.
_ldap._tcp.dc._msdcs.ad.spielwiese.netz. IN      SRV      1 0 389 adcontrol.ad.spielwiese.netz.
_ldap._tcp.gc._msdcs.ad.spielwiese.netz. IN      SRV      1 0 3268 adcontrol.ad.spielwiese.netz.
_ldap._tcp.pdc._msdcs.ad.spielwiese.netz. IN      SRV      1 0 389 adcontrol.ad.spielwiese.netz.
_kerberos._tcp.ad.spielwiese.netz. IN      SRV      1 0 88 adcontrol.ad.spielwiese.netz.
_ldap._tcp.ad.spielwiese.netz. IN      SRV      1 0 389 adcontrol.ad.spielwiese.netz.
_gc._tcp.ad.spielwiese.netz. IN      SRV      1 0 3268 adcontrol.ad.spielwiese.netz.
_kpasswd._tcp.ad.spielwiese.netz. IN      SRV      1 0 464 adcontrol.ad.spielwiese.netz.
_kerberos._udp.ad.spielwiese.netz. IN      SRV      1 0 88 adcontrol.ad.spielwiese.netz.
_kpasswd._udp.ad.spielwiese.netz. IN      SRV      1 0 464 adcontrol.ad.spielwiese.netz.
adcontrol                IN      A          10.1.1.10
adcontrol                IN      TXT       "AD-Server"
gc._msdcs                IN      A          10.1.1.10
sub                       IN      NS        10.1.1.20
```

Anhang B

Visual Basic Skripte

B.1 Das Skript zur Datenübernahme

```
' Im Endausbau soll das Skript eine Liste von zu pflegenden
' NKZ aus einer Datei "lokaleNutzer.txt" lesen, dann aus einer
' Datei "Nutzer.txt" die Einträge der entsprechenden NKZ zu
' suchen und mit dem aktuellen Bestand vergleichen. Bei
' bereits vorhandenen ggf. ein Update vornehmen, bei neuen
' diese Nutzer anlegen. Zur Findung bereits vorhandener NKZ
' wird die Funktion "FindAccount" genutzt. Die Prozedur zum
' Anlegen heisst "BenutzerAnlegen", welche den Nutzer mit den
' Attributen aus der Nutzer-Datei anlegt. Die Prozedur zum
' Updaten "BenutzerAktualisieren" vergleicht erst den
' aktuellen ObjektDN mit dem vorgegebenen und verschiebt ggf.
' das Objekt mittels "MoveObject". Danach werden die gegebenen
' Attribute verglichen und ebenfalls ggf. aktualisiert.
```

```
Const ADS_UF_PASSWD_CANT_CHANGE = &H40
Const ADS_UF_DONT_EXPIRE_PASSWD = &H10000
Const ADS_UF_USE_DES_KEY_ONLY = &H200000
```

```
Const DOMAIN = "DC=ad,DC=spielwiese,DC=netz"
Const NUTZER = "Nutzer.txt"
Const LOKNUTZER = "lokaleNutzer.txt"
```

```
Dim fso1, fso2, f1, f2, Zeile, Feld, User, objSuch
Set fso1 = CreateObject("Scripting.FileSystemObject")
Set f1 = fso1.OpenTextFile (NUTZER,1,0)
Set fso2 = CreateObject("Scripting.FileSystemObject")
Set f2 = fso2.OpenTextFile (LOKNUTZER,1,0)
```

```
Do while not f2.AtEndOfLine
User = f2.readLine
Loop
f2.Close
```

```
Do while not f1.AtEndOfLine
Zeile = f1.readLine
Feld = split(Zeile,":")
NKZ = Feld(0)
If (InStr(User,NKZ) Or User = "all") Then
Nachname = Feld(1)
Vorname = Feld(2)
Titel = Feld(3)
Struktur = Feld(4)
Mail = Feld(5)
Telefon = Feld(6)
Fax = Feld(7)
Raum = Feld(8)
Gruppe = Feld(9)
Set objSuch = FindAccount(NKZ)
If TypeName(objSuch) = "Object" Then
wscript.echo NKZ + " -- gefunden"
Call BenutzerAktualisieren(NKZ,Nachname,Vorname,Titel,Struktur,\
Mail,Telefon,Fax,Raum,Gruppe,objSuch)
Else
wscript.echo NKZ + " -- nicht gefunden"
Call BenutzerAnlegen(NKZ,Nachname,Vorname,Titel,Struktur,Mail,\
Telefon,Fax,Raum,Gruppe)
End If
End If
Loop
f1.Close
Wscript.Quit(0)

Sub BenutzerAktualisieren(NKZ,Nachname,Vorname,Titel,Struktur,\
Mail,Telefon,Fax,Raum,Gruppe,aktObject)
Dim ouo, objKonto

ArrGroup = split(Gruppe,"_")
Anzahl = UBound(ArrGroup) - LBound(ArrGroup)
sOU=""
For i = Anzahl to 0 step -1
sOU=sOU + "OU=" + ArrGroup(i) + ","
Next
sOU = sOU + DOMAIN
fdn = "CN=" + NKZ + "," + sOU
If fdn <> aktObject.distinguishedname Then
call MoveObject(aktObject.distinguishedname, sOU)
Set aktObject = GetObject("LDAP://" & fdn)
```

```
End If
if aktObject.givenName <> Vorname Then
aktObject.Put "givenName", Vorname
aktObject.Put "displayName", Vorname & " " & Nachname
aktObject.Put "description", Vorname & " " & Nachname
End If
if aktObject.sn <> Nachname Then
aktObject.Put "sn", Nachname
aktObject.Put "displayName", Vorname & " " & Nachname
aktObject.Put "description", Vorname & " " & Nachname
End If
if aktObject.title <> Titel Then
if len(Titel) <> 0 Then
aktObject.Put "title", Titel
Else
aktObject.Put "title", CStr(" ")
End If
End If
if aktObject.department <> Struktur Then
if len(Struktur) <> 0 Then
aktObject.Put "department", Struktur
Else
aktObject.Put "department", CStr(" ")
End If
End If
if aktObject.mail <> Mail Then
if len(Mail) <> 0 Then
aktObject.Put "mail", Mail
Else
aktObject.Put "mail", CStr(" ")
End If
End If
if aktObject.telephoneNumber <> Telefon Then
if len(Telefon) <> 0 Then
aktObject.Put "telephoneNumber", Telefon
Else
aktObject.Put "telephoneNumber", CStr(" ")
End If
End If
if aktObject.facsimileTelephoneNumber <> Fax Then
if len(Fax) <> 0 Then
aktObject.Put "facsimileTelephoneNumber", Fax
Else
aktObject.Put "facsimileTelephoneNumber", CStr(" ")
End If
End If
```

```
if aktObject.physicalDeliveryOfficeName <> Raum Then
if len(Raum) <> 0 Then
aktObject.Put "physicalDeliveryOfficeName", Raum
Else
aktObject.Put "physicalDeliveryOfficeName", CStr(" ")
End If
End If
aktObject.SetInfo

End Sub

Sub BenutzerAnlegen(NKZ,Nachname,Vorname,Titel,Struktur,Mail,\
Telefon,Fax,Raum,Gruppe)
Dim ouo, objKonto

ArrGroup = split(Gruppe,"_")
Anzahl = UBound(ArrGroup) - LBound(ArrGroup)
sOU=""
For i = Anzahl to 0 step -1
sOU=sOU + "OU=" + ArrGroup(i) + ","
Next
sOU="LDAP://" + sOU + DOMAIN
wscript.echo sOU
Set ouo = GetObject(sOU)
Set objKonto = ouo.Create("user", "CN="+NKZ)
With objKonto
.Put "sAMAccountName", NKZ
.Put "name", NKZ
.Put "displayName", Vorname & " " & Nachname
.Put "description", Vorname & " " & Nachname
.Put "givenName", Vorname
.Put "sn", Nachname
If len(Titel) <> 0 then
.Put "title", Titel
End If
If len(Struktur) <> 0 then
.Put "department", Struktur
End If
If len(Mail) <> 0 then
.Put "mail", Mail
End If
If len(Telefon) <> 0 then
.Put "telephoneNumber", Telefon
End If
If len(Fax) <> 0 then
```

```
.Put "facsimileTelephoneNumber", Fax
End If
If len(Raum) <> 0 then
.Put "physicalDeliveryOfficeName", Raum
End If
.Put "altSecurityIdentities", "KERBEROS:" & NKZ & \
"@SPIELWIESE.NETZ"
.Put "userPrincipalName", NKZ & "@ad.spielwiese.netz"
.Put "profilePath", "\\Server\daten\profile\" & NKZ
.Put "homeDirectory", "\\Server\daten\home\" & NKZ
.Put "homeDrive", "H:"
'.Put "streetAddress", "Straße der Nationen 62"
'.Put "postalCode", "09107"
'.Put "l", "Chemnitz"
'.Put "c", "DE"
'.Put "co", "Deutschland"
'.Put "st", "Sachsen"
'.Put "company", "TU Chemnitz"
.SetInfo
End With

With objKonto
.SetPassword "EggatPwd_123." 'Hier sollte noch ein\
Passwortgenerator rein!
.AccountDisabled = False
.SetInfo
End With

With objKonto
flag = .Get("userAccountControl")
setflag = ADS_UF_PASSWD_CANT_CHANGE +_
ADS_UF_DONT_EXPIRE_PASSWD + ADS_UF_USE_DES_KEY_ONLY
If (flag AND setflag) = 0 Then
.Put "userAccountControl", flag OR setflag
.SetInfo
End If
End With
End Sub

Function FindAccount(ByVal strName)
path = "LDAP://" + DOMAIN
sql = "SELECT ADsPath FROM '" & path & "' WHERE objectClass='User'\
and name =' " & strName & "'"

Set objconn = CreateObject("ADODB.Connection")
Set objcomm = CreateObject("ADODB.Command")
```

```
objconn.Provider = "ADsDSOObject"
objconn.open = "Active Directory Provider"

Set objcomm.ActiveConnection = objconn

objcomm.CommandText = sql
'objcomm.Properties("PageSize") = 50
objcomm.Properties("Searchscope") = 2

Set rs = objcomm.Execute

If rs.eof Then
Set FindAccount = Nothing
Else
Set FindAccount = GetObject(rs("ADsPath"))
End If
End Function

Sub MoveObject(DNToMove, DNDestination)
Dim oContainer
Set oContainer = GetObject("LDAP://" & DNDestination)
oContainer.MoveHere "LDAP://" & DNToMove, vbNullString
End Sub
```

B.2 Umgebende Text-Dateien

B.2.1 Nutzer.txt

```
mita:Nachname:Vorname:Titel:Abteilung:mail@uni.netz:5311111:\
5312222:1/B001:Mitarbeiter_Mathematik
studi:Name:Vname::Fakultät für Informatik:studi@uni.netz::::\
Studenten_Informatik
NKZ1:NachName:VorName::Fak:Mail:123 3546:::Testnutzer
```

B.2.2 lokaleNutzer.txt

```
# In der Datei wird die letzte Zeile! als kommaseparierte
# Liste von NKZ genommen, welche in das AD übernommen werden
# sollen.
# Besonders wird das Wort "all" behandelt, welches auf alle
# NKZ matcht.
studi,NKZ1,mita
all
```