1 Erste Schritte

1.1 Fähigkeiten und Vorteile

- Samba lässt jeden UNIX-Rechner in der Netzwerkumgebung von Windows erscheinen
- Zugriff auf UNIX-Rechner von Windows aus, genau wie auf jeden anderen Windowsrechner möglich

1.1.1 Fähigkeiten:

- WINS-Server
- Computersuchdienst
- Logon Server
- PDC
- Diagnosewerkzeuge

1.1.2 Vorteile:

- Entfernte Administration
- Zentrale Konfiguration
- Stabilität
- Skalierbarkeit
- Flexibilität
- Offenheit

1.2 Das Server-Message-Block Protokoll

- SMB = Server Message Block
- Client-Server Protokoll, arbeitet nach Frage-Antwort Prinzip
- Client sendet Anfrage an Server, Server schickt Antwort an Client zurück
- Client verwendet Freigabe
- Server stellt Freigabe zur Verfügung

1.3 Eine einfache Konfiguration

Grundsätzlicher Aufbau der smb.conf:

- Gleicht dem Aufbau von INI-Dateien unter Windows
- Datei ist in mehrere Abschnitte unterteilt
- Abschnitte werden jeweils durch Abschnittsnamen eingeleitet
- Abschnittsnamen stehen in []-Klammern
- Jeder Name darf nur einmal vorkommen (Bsp: [public]; [home])
- Inhalte der Abschnitte sind Parameterzuweisungen
- Alle Abschnitte stehen den Clients als Drucker- oder Verzeichnisfreigabe zur Verfügung
- Ausnahme: [global]
  [global] ist keine Freigabe, sondern leitet globale Servereinstellungen ein
- Weitere Freigaben können jederzeit durch anlegen weiterer Abschnitte eingeleitet werden

2 Bestandteile von Samba

2.1 Die Servertools

- Die Dämonen smbd und nmbd sind bei laufendem Samba in der Prozesstabelle sichtbar

2.1.1 Smbd

- Zentraler Serverprozess
- Zuständig für die eigentlichen Datei- und Druckdienste
- Mehrere smbd im System möglich
- Einer dieser Prozesse hört auf TCP Port 139 und nimmt neue Verbindungen entgegen
- Jede neue Verbindung stößt neuen smbd-Prozess an
- Trennen eines Client vom Samba-Server durch Abfrage der Prozessnummer über smbstatus und killen des Prozesses
- Jeder aktive Client braucht ca. 1-2 MB Hauptspeicher

2.1.2 Nmbd

- Zuständig für NetBIOS Namens- und Datagrammdienste
- Reserviert beim Start von Samba die entsprechenden NetBIOS – Namen
- Kann WINS-Server sein
- Zuständig für Computersuchdienst

2.1.3 Sonstige Dienste

- testparm
- smbpasswd
- smbcontrol
- winbindd

2.2 Weitere Serverkomponenten

- Zentrale Konfigurationsdatei = smb.conf
  ➔ als fester Bestandteil des Systems installiert
  ➔ unter */etc/smb.conf* oder */etc/samba/smb.conf*
  ➔ wenn Samba selbst kompiliert: */usr/local/samba/lib/smb.conf*

2.3 Die Clients

- Zugriff als Client auf Windowsrechner im Netz möglich
- Interessant z.B. für Datensicherung
- Vor allem interessant für die Diagnose von Problemen in einer reinen Windows-Umgebung
- Fehlermeldungen der Client-Werkzeuge aussagekräftiger als Windows-Werkzeuge

2.3.1 smbclient

- Zugriff auf Freigaben von NT-Rechnern
- Drucken auf von NT bereitgestellten Rechnern
- NT-Freigaben in tar-Dateien sichern
- Erfragen der Serverliste im Netz

2.3.2 nmblookup

- Diagnose-Werkzeug für die NetBIOS-Namensauflösung
- Findungsversuche zweier Computer die sich nicht finden nachstellbar
- WINS-Server können befragt werden
- NetBIOS Node Status kann abgefragt werden
- nbtstat = entsprechendes Programm unter Windows

3 NetBIOS

!! NetBIOS ist kein Protokoll sondern eine Schnittstelle von Rechnern!!

- Kommunikation untereinander über NetBIOS sobald Windowsrechner Netzlaufwerke verbinden, sich gegenseitig sehen oder Drucker freigeben
- NBT = NetBIOS over TCP/IP

3.1 NetBIOS-Dienste

3.1.1 Namensdienste

- Rechner können sich so gegenseitig im Netz identifizieren
- Namensdienst hat nichts zu tun mit der Anzeige der Netzwerkumgebung
- Computersuchdienst (zuständig für Netzwerkumgebung) abhängig von einem funktionierenden Namensdienst
- Rechner bekommen Namen um im Netz gefunden werden zu können
- Laufende Anwendungen können so nach der Identifizierung miteinander kommunizieren
- Namen bis zu 16 Zeichen lang, jedoch nur 15 Zeichen nutzbar
  16. Byte = Zahl von 0x00 bis 0xff kennzeichnet den Ressourcentyp des Namen
- NetBIOS-Namen existieren im flachen Namensraum, d.h. kein Äquivalent zu Domainbezeichnungen
- a-z, A-Z; 0-9; ! @ # $ % ^ & ( ) – ` { } . ~
- Es können beliebig viele Namen für jede Anwendung reserviert werden
- Unter einem der Namen können Verbindungen aufgebaut und Daten ausgetauscht werden
- Gilt für Clients und Server
- Wollen 2 Anwendungen per NetBIOS kommunizieren, muss erst der Server Bereitschaft kundtun, Verbindungen entgegen zu nehmen
- Dazu muss er seinen Namen im Netz reservieren (Reservierung per Broadcast, alle im Netz können mithören)
  ➔ Rechner kündigt per Broadcast an unter einem bestimmten Namen erreichbar zu sein
  ➔ Gibt es keinen Protest, schickt er die Reservierung ein zweites mal
  ➔ Nach dem dritten erfolgreichen Reservierungsversuch ist der Name endgültig reserviert
- Will ein Client mit einem Server reden braucht er ebenfalls einen eindeutigen im Netz reservierten Namen, Verfahren ist identisch
- Zusätzlich braucht er die MAC-Adresse des Servers

3.2 Datagrammdienst

- Versendete Daten werden in einzelne Pakete aufgeteilt und an den Zielrechner ausgeliefert
- Geht ein Paket verloren, kann man nichts machen
- Aufeinander können folgende Pakete in vertauschter Reihenfolge beim Empfänger ankommen
- Eine Duplizierung der Pakete kann ebenfalls vorkommen, d.h. Pakete können mehrfach ankommen
- Jedoch nur geringer Aufwand beim Daten versenden
- Es kann an mehrere Rechner gleichzeitig gesendet werden

3.2.1 Sitzungsdienst

- Zuverlässiger als Datagrammdienst
- Zwei Rechner vereinbaren eine NetBIOS-Sitzung
- Daten die über diese Verbindung übertragen wurden kommen auf jeden Fall an, in der richtigen Reihenfolge
- Fehlermeldung bei Nichtübertragung

Dienst	**Protokoll**	**Port**	**Samba-Prozess**
• Namensdienst	• UDP	• 137	• nmbd
• Datagrammdienst	• UDP	• 138	• nmbd
• Sitzungsdienst	• TCP	• 139	• smbd

3.3 NetBIOS-Implementationen

3.3.1 NetBEUI

- *„sambasrv"* ➔ MAC-Adresse
  ➔ Client findet Server ausschließlich über Broadcast
  ➔ schickt eine Anfrage über Broadcast wer sich für gesuchten Namen verantwortlich fühlt
  ➔ Server dem der reservierte Name gehört schickt MAC-Adresse aus dem ROM seiner Netzwerkkarte als Antwort
  ➔ Client kann nun Verbindung aufbauen
  ➔ es können hier nur Rechner der gleichen Broadcastdomaine miteinander reden
  ➔ sobald Router im Einsatz kann reines NetBEUI nicht mehr verwendet werden

3.3.2 TCP/IP

- *„sambasrv"* ➔ IP-Adresse ➔ MAC-Adresse
  ➔ Client muss IP-Adresse des Server per Broadcast im Netz herausfinden
  ➔ wurde IP-Adresse herausgefunden kommen die bekannten Mechanismen zum tragen
  ➔ ist der Rechner im eigenen Subnetz wird direkt eine ARP-Anfrage nach der MAC-Adresse ausgelöst
  ➔ Rechner nicht im eigenen Subnetz, wird der entsprechende Router anhand der Routingtabelle herausgefunden und dann dessen MAC-Adresse per ARP festgestellt
  ➔ ARP = Adress Resolution Protocol; 4 Schritte:
  1. Datenpaket wird von der IP an die Ethernet-Netzwerk-Schnittstelle der eigenen Station übergeben. Diese sucht die zugehörige MAC-Adresse in der eigenen temporären Tabelle. Ist ein gültiger Eintrag vorhanden wird ein Ethernet-Paket mit der gefundenen Adresse versehen und abgeschickt.
  2. Kein gültiger Eintrag vorhanden, wird ein ARP-Broadcast-Paket mit der IP-Adresse des Zielhosts erzeugt und anschließend gesendet
  3. Alle Rechner im LAN erhalten das Broadcastpaket und vergleichen die darin enthaltene Ziel-IP-Adresse mit ihrer eigenen. Die Station mit der gesuchten IP-Adresse sendet als Antwort ein ARP-Paket mit der gesuchten Ethernet-Adresse an den Anfragenden Rechner
  4. Host A trägt die empfangene MAC-Adresse von Host B in seine Tabelle ein und sendet das IP-Paket, sowie alle eventuell nachfolgenden direkt an Host B.

!!Broadcast ist nicht Routingfähig!!

4 Namenauflösung per Broadcast

- Vor allem für kleine Netze, funktionieren ohne großen Aufwand
- Basis vieler Netze, skaliert jedoch nicht besonders gut
- *nmblookup* = wichtiges Diagnosewerkzeug für Broadcast Namensauflösung
  ➔ Man kann damit direkt eine Namensanfrage auslösen
  (*nmblookup server; nmblookup smbpc01*)
- *nmblookup* nimmt die Adresse aus der Zeile *„interfaces = „* aus der *smb.conf*
- Unter Windows kann man Namensanfrage nicht isoliert auslösen, man muss dazu eine Verbindung aufbauen
- Windows hat Cache der erfolgreiche Anfragen zwischenspeichert
- Anzeige durch *nbtstat –c*
- Löschen mit *nbtstat –R*
- Mit *nmblookup* (UNIX) und *nbtstat* (Windows) kann man sich zusätzlich die von einem Rechner reservierten Namen ausgeben lassen = Node Status Request
  ➔ eingeleitet durch Parameter *–A <IP-Adresse>*
  ➔ zeigt dass ein Rechner gleich mehrere Namen reserviert

4.1 NetBIOS-Namen

- NetBIOS-Namen in Einzel- und Gruppennamen aufgeteilt, weil bestimmte Dienste eindeutig benannt werden müssen und andere Anwendungen mit mehr als einem Partner gleichzeitig kommunizieren müssen
- Einzelnamen (z.B.- Computername selbst) darf nur ein einziges Mal im gesamten Netz auftauchen, sie werden reserviert und stehen dem entsprechenden Rechner exklusiv zur Verfügung
- Gruppennamen, im Node Status Request durch *<GROUP>* markiert, kann es mehrfach geben, interessant als Ziele für NetBIOS-Datagramme (z.B. reserviert jeder Rechner einer Arbeitsgruppe einen bestimmten Namen, so kann ein Rechner mit einem einzigen verschickten Datagramm sämtliche Rechner der Arbeitsgruppe erreichen)
- Es kann trotzdem durchaus vorkommen dass ein Einzelname mehrfach vorkommt, der kann jedoch durch das 16. Byte eines NetBIOS-Namens eindeutig identifiziert werden

4.1.1 Häufig auftauchende NetBIOS-Einzelnamen

- *Computername<00>:*
  wird von jedem Rechner als NetBIOS-Einzelname reserviert, damit identifiziert er sich eindeutig im Netzwerk und tut so seine Existenz kund; greift ein Rechner auf Ressourcen anderer Rechner zu wird als Clientname dieser Name benutzt
- *Computername<20>:*
  Wird Für Serverdienst reserviert; soll ein Rechner als Datei oder Druckserver angesprochen werden wird eine Verbindung zu diesem Namen aufgebaut; das ist der Name für jeden Rechner der als Server fungiert
- *Benutzer<03>:*
  Nachrichtendienst des Rechners meldet sich so an. Meldungen die mit net send (WinNT) oder winpopup (Win95) verschickt wurden können empfangen werden und am Bildschirm angezeigt werden
- *Arbeitsgruppe<1d>:*
  so genannter Local Master Browser, pflegt die Liste aller Rechner in der Netzwerkumgebung; macht die Dienste und Shares für alle Netzteilnehmer sichtbar, die in einem lokalen Netz angeboten werden
- *Arbeitsgruppe<1b>:*
  definiert den Domain Master Browser, ist über Subnetzgrenzen hinweg für die Netzwerkumgebung zuständig

4.1.2 NetBIOS-Gruppennamen

- *Arbeitsgruppe<00>:*
  hiermit verkündet der Rechner seine Arbeitsgruppenzugehörigkeit; wird von allem Rechnern einer Arbeitsgruppe/Domain registriert; per Datagramm z.B. kann ein net send an alle Rechner einer Arbeitsgruppe geschickt werden
- *Arbeitsgruppe<1c>:*
  jeder Domain Logon Server reserviert diesen Namen; liegt in einer reinen Windowsumgebung immer auf dem PDC; Clients finden ihre Domain Controller über diesen NetBIOS-Namen
- *Arbeitsgruppe<1e>:*
  alle Rechner die Lokaler Master Browser im Netz werden können reservieren diesen Namen; die Wahlen zum Lokal Master Browser werden über diesen Namen abgewickelt
- *.._MSBROWSE_.<01>:*
  alle Lokalen Master Browser registrieren diesen Namen um sich gegenseitig finden zu können

5 Netzwerkumgebung

- Anzeige in der sämtliche Rechner im Netz aufgeführt sind
- Zur Übersichtlichkeit werden Rechner in Arbeitsgruppen aufgeteilt
- Jeder Benutzer kann frei entscheiden in welche Arbeitsgruppe er gehören will
- Gleiches gilt für NT-Domänen

5.1 Begriffe

Klare Trennung der folgenden Begriffe:

- NetBIOS:
  durch NetBIOS sind Rechner im Netz ansprechbar, können verschiedene Dienste anbieten, wie z.B. die Netzwerkumgebung
- Arbeitsgruppe:
  = reine Liste von Rechnern; Hat mit NetBIOS ausschließlich als Transportmedium zu tun; der die Netzwerkumgebung bereitstellende Dienst ist sehr von einem funktionierenden NetBIOS abhängig, insbesondere vom Namensdienst
- Domäne:
  <> Arbeitsgruppe; bezeichnet eine von vielen Rechnern gemeinsam genutzte Benutzerdatenbank (bei Windows gibt es die Einschränkung dass alle Rechner einer Domäne auch in einer Arbeitsgruppe auftauchen müssen, es müssen aber nicht alle Rechner in der Arbeitsgruppe einer Domäne auch die gemeinsame Benutzerdatenbank nutzen)

!! Die Netzwerkumgebung ist mitunter sehr instabil, es kann durchaus vorkommen, dass verfügbare Rechner nie auftauchen oder aber Rechner die nicht mehr verfügbar sind noch lange Zeit in der Netzwerkumgebung zu sehen sind!!!

Zur Vermeidung oben genannter Probleme gibt es den LMB (Local Master Browser) der die Netzwerkumgebung pflegt.

- Wird nicht zentral bestimmt sondern gewählt
- Wahl findet statt wenn einer der beteiligten Rechner feststellt, dass es keinen LMB gibt (Windows kann beispielsweise diese Wahl anstoßen, wird bei WIN 95 die Netzwerkumgebung geöffnet und es erscheint eine Taschenlampe, wird der LMB gesucht)

5.2 Wahl zum Local Master Browser

- Wahl erfolgt über Datagramme an den Gruppennamen *arbeitsgruppe<1e>*
- Der Rechner der das Fehlen des LMB bemerkt schickt ein Datagramm an obigen Gruppennamen
- Jeder Rechner, der diesen Namen reserviert hat hört das Datagramm und entschiedet, wie er selbst vorgehen soll
- Paket verbleibt zusätzlich Broadcast-Paket im lokalen Netz des versendenden Rechners
- Es muss in jedem Subnetz für jede dort anzuzeigende Arbeitsgruppe einen LMB geben
- Drei Kriterien zur Sicherung der Wahl zum LMB:
  1. Übertragung der Betriebssystem-Version des versendenden Rechners. „der beste Rechner gewinnt die Wahl";
     Beispiel:
     ➔ eine Windows NT Workstation empfängt ein Pakt von einem Windows NT Server, sie entscheidet, dass sie die Wahl verloren hat
     ➔ empfängt sie dieses Paket jedoch von einem Windows 95 Rechner hält sie sich für geeigneter den LMB zu übernehmen und versendet selbst ein Wahlpaket mit ihren Parametern; der Windows 95 empfängt dies und sieht dass er verloren hat.
  2. gibt es mehrere Windows NT Workstations im Netz (die Wahl wäre dann unentschieden) kommt die Uptime der Rechner ins Spiel, d.h. der Rechner der am längsten läuft gewinnt die Wahl
  3. Haben zwei Rechner die gleiche Uptime (z.B. nach einem Stromausfall) kommt der NetBIOS-Name des Rechners zum Zug (letztes und eindeutiges Entscheidungskriterium). Der alphabetisch vorn stehende Rechner gewinnt

!! Samba ordnet sich zwischen Windows 95 und Windows NT ein, d.h. gegen Win95 gewinnt Samba, gegen WinNT verliert Samba!!

Zusammenfassung:
Will ein Benutzer die Netzwerkumgebung anschauen kontaktiert sein Rechner den LMB über den NetBIOS-Namen *arbeitsgruppe<1d>*. Genauso finden auch die Server, die angezeigt werden wollen, den LMB als zentrale Stelle, die die Liste der Server pflegt. Gibt es Keinen LMB im Netz, kann die Wahl zum LMB von jedem Teilnehmer im Netz angestoßen werden.
Clients bemerken den Wechsel des LMB oft erst sehr lange nachdem dieser stattgefunden hat. Server müssen sich erst bei dem neuen LMB anmelden bevor eine Liste wieder vollständig ist. Deshalb sollte die Funktion des LMB von einem Rechner übernommen werden der lange läuft. Workstations sind dazu schlecht geeignet. Ein Sambaserver kann mit geeigneten Einstellungen dazu gebracht werden auf jeden Fall die Funktion des LMB zu übernehmen.

!! LMBs gleichen mit dem Domain Master Browser die Namensliste ab!!!

6 NetBIOS über Subgrenzen

- Größere Firmen meist mehr als ein LAN
- Verschiedene Gebäude/Standorte verbunden über Router bzw. jemand wählt sich in das Firmennetz ein
- Namensauflösung funktioniert dann nicht mehr wie beschrieben, bei Namensreservierung und –auflösung ausschließlich über Broadcast können Rechner hinter Routern nicht erreicht werden
- Broadcasts verbleiben in den Subnetzen in denen sie ausgesendet wurden

Zwei Lösungsmöglichkeiten:

6.1 LMHOSTS

- Einfachster Weg, Namensauflösung über Subgrenzen hinweg zu realisieren ➔ statische Tabelle ➔ unter Windows in der Datei *LMHOSTS* (liegt abhängig von der Windowsversion in verschiedenen Verzeichnissen (Suche nach *LMHOSTS.SAM*))
- Ähnlich der Datei */etc/hosts* unter Unix
- Einträge der *LMHOSTS* mit Zusatz *#PRE* ➔ legt fest in welcher Reihenfolge die Namensauflösung vorgenommen wird
- Kein *#PRE* vorhanden, wird zunächst eine konventionelle Namensauflösung per Broadcast versucht; Schlägt diese fehl ➔ wird in *LMHOSTS* nachgesehen
- Mit Zusatz wird direkt in der Wert der *LMHOSTS* verwendet
- Nachteile: - muss auf jedem Rechner gepflegt werden
  - macht diese Art der Namenspflege sehr schnell unwartbar
- Zentrale *LMHOSTS* durch Statement *#INCLUDE* (man stellt an zentraler Stelle eine Freigabe zur Verfügung in der *LMHOSTS* steht und fügt sie automatisch bei jedem booten in die Liste auf den Clients ein. Einmaliges aufsetzen der *LMHOSTS*:
  BSP: *192.168.1.1 samba #PRE*
  #INCLUDE ||samba\public\lmhosts
  #PRE und #INCLUDE immer Groß

6.2 WINS

- Zentraler Server der die NetBIOS-Namen in einer Datenbank dynamisch pflegt
- Jede NetBIOS-Applikation meldet sich im Netz mit ihrem Namen an
- IP-Adresse dieses Servers muss jedem Rechner mitgeteilt werden, bei Windows in den Eigenschaften des TCP/IP-Protokolls im Reiter WINS-Adresse
- Setzt man den DCHP-Server ein kann ebenfalls der WINS-Server festgelegt werden
- Samba bekommt die Adresse mit dem Parameter: *wins server = <ip-adresse>* im Abschnitt *[global]* der *smb.conf* mitgeteilt
- Kennt ein Client die IP-Adresse des WINS-Server ist es egal ob er sich im gleichen Subnetz befindet oder nicht
- Namensreservierung über UDP-Paket an den WINS-Server
- Gerichtete Pakete leitet der Router an den WINS-Server weiter wie jedes andere Paket; dieser prüft in seiner Tabelle ob der Name reserviert ist
- ist das nicht der Fall wird spontan eine Bestätigung der Reservierung geschickt, die Reservierung gilt nur eine begrenzte Zeit und muss rechtzeitig erneuert werden.
- Ist der Name bereits reserviert befragt der WINS-Server den bisherigen Besitzer ob der Name noch benötigt wird; will er den Namen noch verwenden, bekommt der Anfragende eine Ablehnung; benötigt der alte Besitzer den Namen nicht mehr wird dem Anfragenden eine positive Antwort geschickt
- Dadurch ist ein spontanes Booten eines Rechners nach einem Absturz möglich
- Konfiguration von Samba als WINS-Server durch den Parameter: *wins support = yes*
- Durch diesen Parameter kann Samba nach einem Neustart bei allen Clients und sonstigen Servern als WINS-Server eingetragen werden. Werden diese neu gestartet melden sie sich beim WINS-Server an

6.3 Vorteile von WINS

- Wartezeiten bei der Namensreservierung entfallen, da hier nur ein einziger Rechner sämtliche reservierte Namen registriert und sie in seiner Tabelle nachschauen kann (bei Broadcast muss eine Namensreservierung 3 mal gesendet werden um einen Namen endgültig reservieren zu können)
  ➔ Namensreservierung per WINS deutlich schneller und weniger netzbelastend
  (➔ Einsatz eines WINS-Server sollte auch in Erwägung gezogen werden bei nur einem einzigen Subnetz)

Anmerkung
- Netzwerkweit darf es nur einen WINS-Server geben, auch bei mehreren Arbeitsgruppen, Domänen
- Setzt man mehrere ein, hat man getrennte Namensräume und bekommt massive Probleme da Windows Namen sowohl per WINS als auch per Broadcast auflöst
- Rechner im einen Namensraum können mit Rechner, die an einen WINS-Server angeschlossen sind, nicht kommunizieren, da die Namen nicht aufgelöst werden können
- Namen die unter WINS nicht bekannt sind werden von Windows zusätzlich per Broadcast aufgelöst (d.h. man findet einige Rechner nur per WINS andere auch lokal)
  ➔ dadurch wird die Fehlerdiagnose stark erschwert
- Unter Windows NT kann man jedoch mehrer WINS-Server einsetzen, die sich gegenseitig abstimmen
- Diese Applikation stellt sicher dass unabhängig von der Anzahl der WINS-Server nur eine Namensdatenbank von den Clients gesehen wird
- WINS-Server stellen sich so den Clients als eine konsistente Datenbank dar
- Erst ab Version Samba 3.0

7 Windows-Namensauflösung im Detail

Zusammenfassung der Mechanismen mit denen ein Windows-Rechner einen Namen in eine IP-Adresse auflöst:

Zwei Arten von Anwendungen:
- NetBIOS-Anwendungen:
  Klassische Windows-Programme um z.B. ein Laufwerk mit einem Server zu verbinden, Outlook mit dem Exchange-Server zu verbinden; ebenso wie die gesamte Netzwerkumgebung
- TCP/IP-Anwendungen:
  z.B. TelNet, ping, Netscape gibt es nur in der TCP/IP-Protokollfamilie

7.1 NetBIOS Anwendungen

Will eine NetBIOS-Anwendung einen Namen auflösen, geschieht das in mehreren Schritten, die nacheinander ausgeführt werden, bis der Name gefunden ist:
1. System schaut im NetBIOS-Namenscache nach (kann durch nbtstat –c vom Benutzer abgefragt werden)
2. WINS-Server (soweit vorhanden) wird befragt
3. Broadcastanfrage, wenn durch WINS der Name nicht aufgelöst werden konnte
4. In LMHOSTS nachgesehen
5. Übergabe an das Auflösungssystem für TCP/IP-Anwendungen (soweit in den Eigenschaften von TCP/IP die DNS-Auflösung für NetBIOS aktiviert ist)

- Trägt man Namen in die Datei *LMHOSTS* einträgt werden diese erst nach den WINS- und Broadcast-Timeouts berücksichtigt
- Mit dem Zusatz *#PRE* können diese sofort auflösen, sie werden dann beim Neustart dauerhaft in den NetBIOS-Namenscache geladen
- Im laufenden Betrieb kann durch ein *nbtstat –R* das Laden in den Namenscache erzwungen werden
- Durch die IP-Adressvergabe DHCP kann man Windows-Servern die IP-Adresse des WINS-Server mitteilen

7.2 TCP/IP-Anwendungen

- Namensauflösung einfacher als bei NetBIOS-Anwendungen

1. In Datei HOSTS nachgesehen
2. Befragung des DNS-Server (soweit konfiguriert)
3. DNS-Name wird so wie er ist an die NetBIOS Namensauflösung übergeben (für interne Systeme kann somit vermieden werden sie ins DNS aufnehmen zu müssen;
   Beispiel:
   Will man einen Proxy unter dem Namen proxy einrichten, reicht es auf dieser Maschine einen korrekt konfigurierten nmbd zu installieren der den Namen proxy registriert, so kann man auf allen Browsern einfach proxy eintragen)

7.3 Samba als Client

- Samba-Namensauflösung weniger kritisch als die von Windows-Systemen, da Samba in der Regel nur als Server auftritt
- Samba als Server ist es gleichgültig wie Namen aufgelöst werden können

Zwei Situationen in denen Samba auflösen muss:
1. Smbclient:
   Samba als Client muss offensichtlich Namen auflösen
2. Samba als Domänenmitglied:
   Mit Parameter password server wird Samba als Domänenmitglied mitgeteilt, welcher Domänencontroller für Passwörter zuständig ist. (Hier ist es enorm wichtig, dass für diese Funktion die Namensauflösung korrekt funktioniert)

- Samba kennt 4 Mechanismen zur Namensauflösung: *Broadcast, WINS, LMHOSTS und die normale Unix-Namensauflösung*
- Durch Parameter *resolve order* wird die Reihenfolge der Mechanismen zur Namensauflösung festgelegt (mit den vier Werten *bcast, wins, lmhosts und hosts*)
- Standardreihenfolge: name resolve order = lmhosts host wins bcast
- Es kann von Vorteil sein die Reihenfolge auf: *name resolve order = lmhosts wins bcast host* festzulegen oder vollständig auf die DNS-Namensauflösung zu verzichten, da es zu dem Problem kommen kann dass man auf einen DNS-Timeout warten muss bevor die Windows-Namesauflösung benutzt wird

8 Browsing über Subnetzgrenzen hinweg

- Dienst: Domain Master Browser ist zu installieren wenn eine einheitliche Arbeitsgruppe über Subnetzgrenzen hinweg gewünscht ist
- Sammelt die Serverlisten von allen LMBs und gibt sie auf Anforderung wieder heraus
- DMB ist passiv, wartet bis sich ein LMB mit ihm synchronisieren will
- Aufgabe der LMBs sich regelmäßig danach zu erkundigen wo der DMB sitzt um dann die Serverlisten abzugleichen
- Parameter *domain master = yes* in der *[global]* Sektion, damit ein Samba-Server die Aufgabe eines DMB übernehmen kann
- Nmbd versucht dann diese Funktion im Netz zu übernehmen
- Verwendet speziellen Namen *arbeitsgruppe<1b>*, der den Samba-Server als DMB im Netz identifiziert
- Der DMB übernimmt gewöhnlich auch die Rolle des LMB in seinem Lokalen Netz
- Setzt man Samba in einer NT-Domäne ein sollte man dem Primary Domain Controller die Rolle des DMB überlassen, wenn ein solcher vorhanden ist, da der PDC diesen speziellen Namen für sich reserviert und man ihn nicht bewegen kann diese Rolle nicht zu übernehmen. Dadurch kann das sog. Cross-Subnet-Browsing u.U. nicht mehr funktionieren

8.1 Browsing mit vielen Arbeitsgruppen

Wie bekommt der LMB die Liste der Arbeitsgruppen, welche direkt von ihm vorgehalten wird? (Sie wird unter Anderem in der Netzwerkumgebung unter Microsoft Windows Netzwerk als Liste sämtlicher Arbeitsgruppen angezeigt)

- Jeder LMB reserviert sich einen speziellen Gruppennamen: *.._MSBROWSE_.<01>*
  (Die Punkte stehen für die Ascii-Werte eins und zwei)
- Der LMB sendet regelmäßig seine Existenz an diesen Gruppennamen
- Alle anderen LMBs sammeln diese Ankündigungen um den Clients die Liste der vorhandenen Arbeitsgruppen und LMBs mitteilen zu können.
- Kommen DMBs ins Spiel wird das ganze etwas komplizierter
- Samba fragt den WINS-Server regelmäßig nach allen DMBs
- DMBs werden in zufälligen Abständen kontaktiert um die Browserlisten mit ihnen abzugleichen
- Durch Parameter *enhanced browsing = no* lässt sich verhindern dass Arbeitsgruppen die nicht mehr existieren weiterhin in der Netzwerkumgebung auftauchen und sich nicht löschen lassen

Warum könnte man virtuelle Samba-Server brauchen?

- Zur Selbstkonsolidierung, kann es notwendig sein unter mehreren Namen im Netz aufzutauchen
- Mit Parameter *netbios aliases* möglich
- Soll der Server jedoch in mehreren Arbeitsgruppen auftauchen muss man den *nmbd* und den *smbd* mit unterschiedlichen Konfigurationsdateien starten

9 SMB-Sitzungen

Es müssen folgende 6 Schritte durchlaufen werden, bevor ein Client auf eine Freigabe zugreifen kann:

9.1 NetBIOS-Namensauflösung

- Um eine Verbindung zum Server aufzubauen muss der Benutzerden Name des Servers herausfinden (z.B. über Doppelklick in der Netzwerkumgebung, über Start ➔ Ausführen, über Netzlaufwerk verbinden im Explorer,…)
- Danach muss der Client die MAC-Adresse herausbekommen, siehe Seite 12 ff

9.2 TCP-Verbindung

- TCP-Verbindung zum Port 139 des Server, wenn Adresse klar ist zu der verbunden werden soll
- *Netstat* (unter Windows und UNIX) um vorhandene TCP-Verbindungen anzuzeigen
- *telnet <ip> 139* um zu prüfen ob die TCP-Verbindung klappt
- Fehlermeldung wird angezeigt wenn Verbindung nicht klappt
- Klappt die Verbindung sieht man über netstat ob die Verbindung ESTABLISHED ist

9.3 NetBIOS-Sitzung

- Mehrere Anwendungen auf einen Server-Rechner, die Namen für sich reserviert haben
- Alle über IP-Adresse des Rechners und dem TCP-Protokoll auf Port 139 erreichbar
- Anhand des TCP-Verbindungsaufbau ist klar welche Serverapplikation angesprochen werden soll ➔ Unterscheidung durch den Servernamen der in der TCP-Verbindung als erstes übertragen wird

9.4 Negotiate Protocol

- NetBIOS-Sitzung aufgebaut und Daten können übermittelt werden
- Innerhalb der NetBIOS-Sitzung wird SMB-Sitzung schrittweise aufgebaut
- Erste Anfrage des Clients an den Server ist ein Negotiate Protocol Request
- In dieser Anfrage wird eine Liste der Protokollvarianten mitgeschickt die der Client beherrscht
- Server wählt ein Protokoll der Liste aus und schickt eine Antwort mit dem Index des Protokolls, welches dann für die Kommunikation genutzt wird
- Mit Parameter *protocol* kann das höchste Protokoll festgelegt werden mit dem Samba arbeiten soll
- Zusätzlich werden in der Antwort zwei weitere Einstellungen verschickt, die Teile des weiteren Ablaufs festlegen
- Server entscheidet ob er die Zugriffssteuerung auf Benutzer- oder Freigabeebene regeln möchte, d.h. zu welchem Zeitpunkt der Benutzer ein Passwort eingeben muss (direkt beim Session SetUp oder erst beim Tree Connect)
- Parameter *security = share* Zugriffsteuerung auf Freigabeebene, *security = user* auf Benutzerebene
- Vorgabe an den Client ob Klartextpasswörter verwendet werden sollen oder verschlüsselte
- Bei Verschlüsselung wird zusätzlich die Herausforderung für das Challenge Response Verfahren mitgeschickt
- Es ist nicht möglich für einige Benutzer Klartextpasswörter zu verwenden und für andere Verschlüsselte

9.5 Session Setup

- Verschicken eines Session Setup mit Benutzernamen vom Client an den Server, nach dem Protokollversion ausgehandelt
- Wurde vom Server im Negotiate Protocol *security = user* verlangt wird das Passwort ebenfalls mitgeschickt (Server kann so die Identität des Benutzers feststellen)
- *Security = share* ➔ Server ignoriert evtl. mitgeschicktes Passwort

9.6 Tree Connect

- Client legt fest welche Freigabe er ansprechen will
- *Security = share* ➔ Server überprüft das Passwort

10 Freigaben und Rechte an Freigaben

- Wichtigster Dienst von Samba ➔ Bereitstellung von Festplattenbereichen für Clients
  ➔ Freigabe

10.1 Rechte an Freigaben

!! Standardmäßig wird immer Leserecht vergeben!! Schreibrecht muss gesetzt werden!!

- Entscheidung zwischen Schreib- und Lesezugriff an Freigaben (bei NT können darüber hinaus auch Vergabe von detaillierten Rechte)
- *security = user* ➔ Server hat die Möglichkeit Zugriff zu geben oder zu verweigern
- keine Parameter angegeben ➔ korrekt angemeldeter Benutzer hat Leserecht
- Vergabe von Rechten an einzelnen Benutzer oder ganze UNIX-Gruppen

Die drei häufigsten Fälle:

- Gleicher Zugriff für alle Benutzer:
  alle Benutzer die sich mit Namen und Passwort am Server anmelden haben Leserecht auf die Freigabe
- Einige Benutzer haben gleichen Zugriff:
  ➔ über die Liste valid users werden die Rechte für einzelne Benutzer vergeben
  ➔ ganze UNIX-Gruppen bekommen über ein vorangestelltes @ Freigaben
- Einige Benutzer haben Leserecht, andere Schreibrecht:
  ➔ alle Benutzer die Zugriff auf die Freigabe bekommen zuerst nur Leserecht
  ➔ Benutzer die auch Schreibrecht haben sollen, bekommen nur zur Standardeinstellung auch ein Schreibrecht

11 Zugriffsrechte im Dateisystem

Vier verschiedene Attribute für Dateien unter DOS:

- Read-Only:
  Inhalt kann nur gelesen aber nicht geschrieben werden. Dateien können nicht gelöscht werden
- System:
  Für spezielle Betriebssystemzwecke
- Hidden:
  Datei wird mit Kommando Dir nicht angezeigt. (versteckte Datei)
- Archiv:
  Wird bei jedem Schreibzugriff gesetzt Backupprogrammen ist es freigestellt dieses Bit zu setzten. Dadurch ist eine inkrementelle Sicherung möglich

!!Diese Bits können unter DOS von jedem Benutzer frei gesetzt werden und auch wieder zurückgesetzt werden!! Schreibschutz somit kein echter Zugriffschutz sondern nur kleine Hilfestellung gegen Fehlbedienungen!!

11.1 Abbildung DOS-Attribute zu Unix-Rechten

- Behandlung Dateiberechtigung größter Unterschied zwischen DOS und Unix-Rechnern

UNIX:
- Führt mit jeder Datei einen Satz Zugriffsrechte
- Aufgeteilt in drei Gruppen von Benutzern: Dateibesitzer, besitzende Gruppe, alle anderen
- 3 Rechte: Lesen, Schreiben, Ausführen

WINDOWS:
- 4 Bits (schreibgeschützt, System, Archiv, versteckt)
- nicht getrennt einstellbar
- kein Bit für Kennzeichnung ausführbarer Dateien
- erkennt ausführbare Dateien an Dateinamenserweiterungen: EXE, COM, CMD, BAT

- Samba darf die drei Bits für Ausführbare Dateien unter Unix nicht seinen DOS-Clients mitteilen

- *Map archive, map system, map hidden* ordnen die Bits Archiv, System, versteckt den Ausführungsbits für Besitzer, Gruppe und alle anderen zu
- -rwxrwxrwx
- schreibgeschützt ➔ alle w's raus
- x bei Besitzer ➔ Archive
- x bei Gruppe ➔ System
- x bei allen anderen ➔ versteckt

	R	W	X
U	4	2	1
G	4	2	1
O	4	2	1

11.2 Erstellungsmasken

- Wird eine Datei neu erstellt übergibt der Client dem Server die Dos-Attribute mit denen er die Datei erstellt haben möchte ➔ Samba formt Unix-Zugriffsrechte
- Rechte die in der *Create Mask* gesetzt sind können möglicherweise in der neuen Datei/Verzeichnis auftauchen (logisches UND)
- Explizite Bitsetzung durch *force create mode* (logisches ODER)

12 Opportunistic Locks – OpLocks

- Mechanismus mit dem einem Client das Cachen von Dateiinhalten erlaubt wird
- Client bekommt Datei solange exklusiv bis Server ihn auffordert die Änderungen zurückzuschreiben und die Sperre freizugeben
- Client A will Datei öffnen und beantragt OpLock auf die Datei
- Gewährt Server dieses OpLock = Zusage dass keiner sonst auf die Datei zugreift
- Client muss so weder bei Lesezugriff Server befragen noch den Schreibzugriff unverzüglich an den Rechner liefern
- 30-40% mehr Geschwindigkeit bei typischen Applikationen eines WindowClient
- Will ein zweiter Client die Datei öffnen schickt der Server dem Client ein OpLock-Break (Anweisung sämtliche lokale Änderungen zurückzuschreiben und den Schreibcache auf dieser Datei auszuschalten
- Erst nachdem A alle Änderungen zurückgeschrieben hat kann B die Datei öffnen
- Da keiner von beiden noch ein OpLock bekommt sehen beide die Änderungen sofort
- Funktioniert innerhalb von Samba
- Greifen UNIX-Prozesse ebenfalls auf Dateien zu gibt es Probleme mit den OpLocks
  ➔ keine vernünftige Dateisicherung mehr weil Clients möglicherweise nicht alle Änderungen zurückgeschickt haben

Lösungsmöglichkeiten:
- Keine OpLocks:
  ➔ massive Performanceprobleme
- Keine OpLocks für einzelne Dateien
- Kernel OpLocks:
  ➔ Problem: Samba kann vom Kernel nicht informiert werden wenn Unix-Prozess auf Datei zugreifen will

13 Verschlüsselte Passwörter

Benutzerauthentifizierung muss 2 Dinge leisten:
- Benutzer muss beweisen dass er das Passwort kennt
- Authentifizierungsprotokoll kann ermöglichen dass das Passwort nicht übertragen werden muss ➔ Zuhörer kann mit den empfangenen Daten nichts anfangen

- Challenge-Response Verfahren

13.1 Symmetrische Verschlüsselung (Samba)

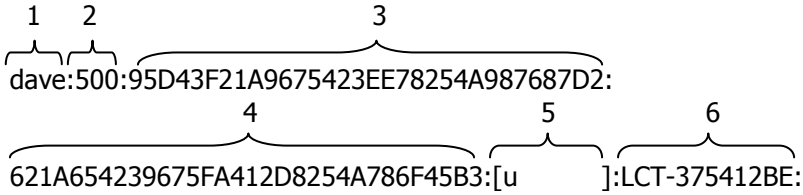
- Bei Übermittlung geheimer Nachrichten welche Dritte nicht lesen können sollen
- Nachricht wird so zerstückelt, dass niemand sie mehr lesen kann, außer das Verschlüsselungsverfahren ist bekannt
- Jedes Verschlüsselungsverfahren hat ein Gegenstück welches die zerstückelte Nachricht wiederherstellt

13.2 Challenge-Response-Verfahren

- Trickreicher Einsatz der Symmetrischen Verschlüsselung
- Damit der Client eine Verbindung zum Server aufbauen kann muss der Benutzer erst seinen Namen und sein Passwort angeben
- Danach baut der Client die Verbindung zum Server auf
- In der Antwort auf diese Anfrage des Clients (der Negotiate Protocol Response) schickt der Server eine Zufallszahl; die sog. Herausforderung
- Client verfügt somit zu diesem Zeitpunkt über drei Werte: Benutzer, Passwort und Herausforderung
- Passwort soll jetzt verschlüsselt über das Netz übertragen werden
- Naiver Ansatz: Herausforderung = Schlüssel (für ein Symmetrisches Verschlüsselungsverfahren)
  Problem: Jeder Zuschauer würde den Schlüssel ebenfalls kennen und so das Passwort entschlüsseln können
- Besser:
  Passwort wird als Schlüssel benutzt um die Herausforderung zu Verschlüsseln
- Die so verschlüsselte Herausforderung wird vom Client im Session Setup zusammen mit dem Benutzernamen an den Server geschickt
- Hat der Server den Session Setup erhalten hat er sich die Zufallszahl gemerkt und hat den Benutzernamen vom Client erhalten
- Aus der Benutzerdatenbank kann er das Passwort des Benutzers ablesen
- Mit dem Passwort als Schlüssel entschlüsselt der Server die verschlüsselte Herausforderung und prüft ob wieder die versendete Zufallszahl herauskommt.
- Ist dies der Fall stimmen die beiden Schlüssel überein
- Stimmt der entschlüsselte Wert nicht mit der gesendeten Zufallszahl überein, wurde zur Verschlüsselung ein anderer Schlüssel (Passwort) benutzt als für die Entschlüsselung; das am Client eingegebene Passwort stimmt also nicht mit dem Passwort überein welches der Server in seiner Benutzerdatenbank gespeichert hat

13.3 Die Datei smbpasswd

- Struktur der smbpasswd:
  Beispiel:



1. Username:
   Name des Benutzers dieses Kontos. Aus Systemkennwortdatei übernommen
2. UID:
   Benutzeridentifikation des Kontos. Aus Systemkennwortdatei übernommen und muss zum Benutzernamen gehören
3. LAN Manager Password Hash
4. NT Password Hash
5. Account Flags:
   besteht aus 11 Zeichen innerhalb eckiger Klammern. Jedes der folgenden Zeichen kann in beliebiger Reihenfolge auftreten, Rest sollte aus Leerzeichen bestehen:
   - U = Konto ist gewöhnliches Benutzerkonto
   - D = Konto momentan deaktiviert, Samba lässt keine Anmeldungen zu
   - N = Konto besitzt kein Passwort
   - W = Vertrauensstellung zu einer Arbeitsstation, die es Samba ermöglicht, als PDC (Primary Domain Controller) zu arbeiten. Dabei können Windows NT Rechner der Domäne beitreten
6. Last Change Time:
   Besteht aus den Zeichen LCT gefolgt von der hexadezimalen Darstellung der Sekunden, die seit Mitternacht des 01.01.1970 verstrichen sind. Gibt an wann der Eintrag zuletzt geändert wurde.

14 Druckfreigaben

!! Drucker werden unter Samba genauso freigegeben wie ein Verzeichnis!!
!! Um diese zur Verfügung zu stellen, müssen diese von Unix aus ansprechbar sein!!

15 Benutzerverwaltung

15.1 Peer-to-Peer-Netzwerke

!! Arbeitsgruppe = Ansammlung von Rechnern (Jeder hat seine eigene Benutzerdatenbank)!!
- Beim installieren einer Arbeitsgruppe bekommt man komplett getrennte Benutzerdatenbanken auf den einzelnen Rechnern
- Erstellt man eine Freigabe auf einem Server und will für diese Rechte vergeben, müssen zunächst die Benutzer eingerichtet werden, die Rechte für diesen Rechnern bekommen sollen
- Transparentes Anmelden: (beim Zugreifen eines Benutzers einer anderen Workstation) Anmeldeversuch mit dem lokal angemeldeten Benutzer und seinem Passwort (dadurch sieht es so aus als ob man nur ein Benutzerkonto verwenden würde)
- Mit dem Benutzermanager für Domänen kann die Administration der Benutzerdatenbanken komplett von einem zentralen Rechner aus erfolgen

15.2 Windows NT Domänen

- Diese Art der Administration skaliert nicht besonders gut
- Jeden Benutzer muss er auf jedem Server geben
- Die lokalen Workstations brauchen ebenfalls separat gepflegte Benutzer
- Lösung des Problems: Einführung des Domänenkonzepts mit WIN NT
- Benutzer bekommen zentrales Konto, welches auf allen Domänenmitgliedern gültig ist
- Domäne realisiert auf PDC; PDC stellt seine Benutzerdatenbank für andere im Netz zur Verfügung
- Alle Domainmitglieder importieren diese Benutzerdatenbank
- Domainmitglieder haben somit 2 gültige Benutzerdatenbanken: die lokale und die des PDC

15.3 Benutzerdatenbanken und SIDs

Unterschiede zwischen UNIX und Windows NT Benutzer:

UNIX:

- Benutzer besteht im wesentlichen aus einer numerischen UserID
- Das Programm login muss beim Anmelden des Benutzers anhand dessen Namen herausfinden welche numerische UserID er hat
- Nachsehen in der Datei */etc/passwd*
- Login prüft das Passwort mit der Datei */etc/shadow*
- Passwort korrekt ➔ Umschaltung in gefundene UserID und start der LoginShell des Benutzers
- Danach ist für UNIX nur noch der numerische Wert von Interesse
- An jedem Prozess hängt damit eine eindeutige Identifikation der Rechte die er hat
- UserIDs gelten nur auf dem Rechner auf dem sie zugeordnet wurden

Windows NT:

- Zunächst wie bei UNIX
- Numerische USerID vorhanden
- Name des Benutzers ist nur während der Anmeldung für das System interessant
- Nach der Anmeldung nur noch die UserID relevant
- UserID jedoch über Rechnergrenzen hinweg gültig
- Deshalb wird der Rechner nicht durch eine kleine Zahl beschrieben sondern durch einen so genannten Security Identifier (SID)
- SID besteht aus zwei wesentlichen Teilen
  erster Teil: 96 Bit lange Zahl, welche die Benutzerdatenbank des SID eindeutig identifiziert
  zweiter Teil: Relative Identifier (RID) ➔ vergleichbar mit der UserID unter UNIX
- USerID jedoch nur zusammen mit den 96 Bit der Benutzerdatenbank verwendet ➔ Benutzer unterschiedlicher Maschinen oder Domänen sind so unterscheidbar