Inhalt

PowerShell

- **Objektorientiert**: Objekte haben Attribute und Methoden
- Nachfolger der CMD
- CMDlets Aufbau: **Verb-Substantiv**
- Variablen durch **vorgestelltes $**-Zeichen
- Mit Pipe (|) kann das Ergebnis eines CMDlets als Eingabe an das nächste übergeben werden

Execution-Policies

- **Restricted**: Skripte werden nicht ausgeführt
- **RemoteSigned**: Nur lokale Skripte und signierte Skripte werden ausgeführt
- **AllSigned**: Nur signierte Skripte werden ausgeführt
- **Unrestricted**: Alle Skripte werden ausgeführt

Parameterarten

- **Benannt**: Angabe mit Namen und vorangestelltem „-„(z.B. -Path)
- **Switch**: Boolean-Input allein durch die Nennung des Parameters (z.B. -Force)
- **Position**: Verwendung des Inputs abhängig von der Position des Inputs (z.B. Copy-Item <Quelle> <Ziel>)

Assoziative Arrays

Ähnlich wie Objekte in JavaScript – Fördern die Übersichtlichkeit und Lesbarkeit

ForEach-Schleifen

- Iteriert über eine vorgegebene Liste
- Keine Zählvariable, sondern eine **Schleifenvariable**
- Schleifenvariable enthält aktuellen Datensatz des Durchlaufs
- Bsp: ForEach($File in $FileList){…}

CSV-Dateien

- **Comma-separated-values**: Trennung durch definierte Trennzeichen (z.B. Komma)
- Die 1. Zeile enthält die Namen der Datenfelder

Verwendung in PowerShell

- **Import**: Import-Csv (z.B. $csvfile = Import-Csv .\File.csv -Delimiter “;”)
- **Export**: Export-Csv (z.B. Get-ChildItem . | Export-Csv -Path file.csv -Delimiter “;”)

Vergleichsoperator “like”

- Vergleicht String mit einem vorgegebenen Muster
- Muster kann Platzhalter enthalten
- Platzhalter:
    - **Stern** (*): Beliebige und beliebig viele Zeichen
    - **Fragezeichen** (?): Ein beliebiges Zeichen
    - **Range** ([q-w]): Ein beliebiges Zeichen aus einer Auswahl an Zeichen

Regular Expressions

➔ Siehe Cheat-Sheet

Active Directory (AD)

- **Verzeichnisdienst** in Windows-Netzen
- Aufbau in Datenbank (NTDS.dit)
- Zur Abfrage der Datenbank wird **LDAP** (Lightweight Directory Access Protocol) verwendet
- In der Struktur vorhanden:
    - **Objekte**: Benutzer, Gruppen, Computer
    - **Orte**: Organistationseinheiten (OU), Domäne
- Jedes Objekt in der Datenbank wird durch den **Distinguished Name** gekennzeichnet (entspricht dem absoluten Pfad in der Baumstruktur; ähnl. Primary-Key)
- Struktur eines Unternehmens wird im AD abgebildet

Hauptkomponenten

- **LDAP**: Protokoll für den Zugriff auf den Verzeichnisdienst, in dem Informationen über Nutzer, Gruppen, Computer und anderen Objekten abgelegt sind
- **Kerberos-Protokoll**: Authentifizierung von Benutzern
- **CIFS/SMB** (Common Internet File System / Server Message Block): Ablage von Dateien im Netzwerk
- **DNS** (Domain Name System): Namensauflösung (früher NetBIOS/WINS)

Bestandteile des Ads

Schema

- Definiert Objekttypen, Klassen, Attribute und Attributsyntax
- Wichtige Klassen: User, Computer, OU, Group

Konfiguration

- Beschreibt die Gesamtstruktur (besteht aus Domänen) und deren Bäume

Domäne

- Beinhält alle Informationen über die Objekte einer Domäne
- Informationen werden im globalen Katalog gespeichert
- Zentral verwaltbarer Sicherheitsbereich
- Wird erstellt durch Installation eines Domänencontrollers
- Domänencontroller speichert sämtliche Objekte einer Domäne (immer nur eine Domäne)

Struktur/Tree

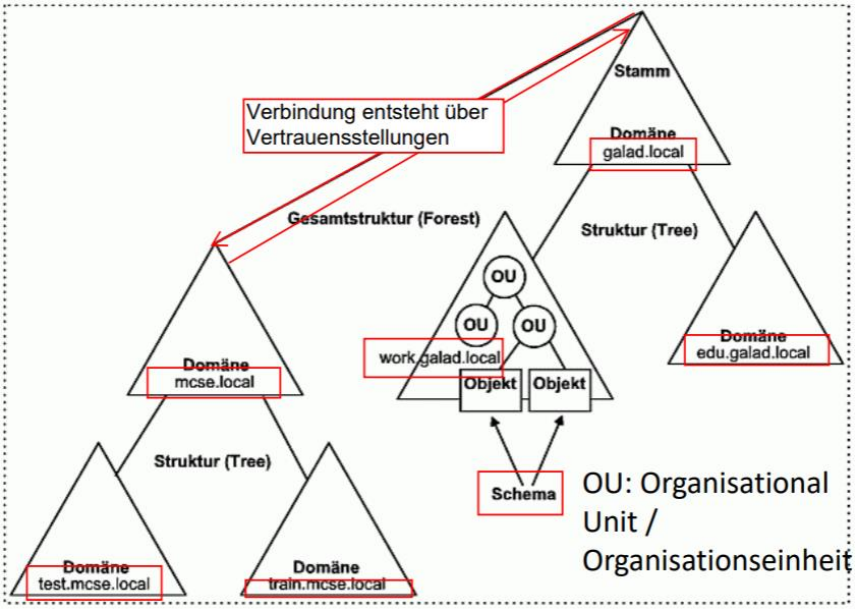
- Entsteht wenn Subdomänen erstellt werden

Gesamtstruktur/Forest

- Zwei Strukturen, die sich im selben AD befinden (braucht zweiten DC)

Vertrauensstellung

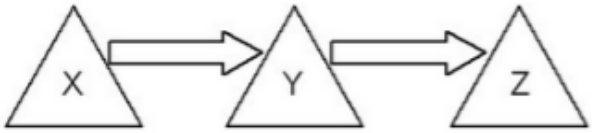
- Kann zwischen zwei oder mehr Domänen aufgebaut werden
- Ermöglicht es Benutzern einer Domäne auf Ressourcen einer anderen zuzugreifen
- Vertrauende Domäne lässt Authentifizierungen der vertrauten Domäne zu



Arten

Unidirektional

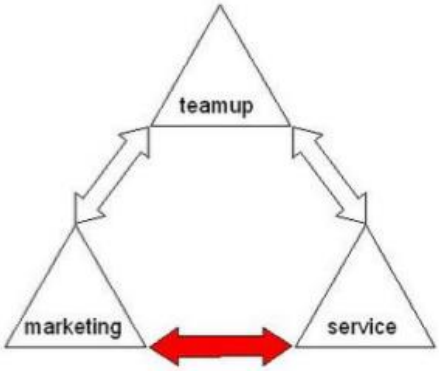
- Nur in eine Richtung
- Nicht durchlässig (Nicht transitiv)
- *X vertraut Y, Y vertraut Z*



-> Y lässt Anmeldungen aus Z zu

Bidirektional

- Beide Richtungen
- Durchlässig (transitiv)
- *Teamup vertraut marketing und service*
    - *Dadurch vertraut service automatisch marketing*



LDAP: Lightweight Directory Access Protocol

- Netzwerkprotokoll zum Zugriff auf Verzeichnisdienste
- Form einer Baumstruktur (directory information tree => dit)
- Arbeitet Objektorientiert: Alle Daten in Objekten, die einer Objektklasse zugeordnet sind
    - Objekte haben Attribute und Methoden

Vererbung bei Objekten

- Neue Klassen können von einer Elternklasse abgeleitet werden
    - Sie erhalten alle Attribute der Elternklasse
    - Kann zusätzliche Attribute haben
- Alle Klassen im AD sind von der Klasse **top** abgeleitet
- Zu jeder Klasse kann ein Objekt angelegt werden und in der Baumstruktur abgelegt werden
- Wurzel des Baumes ist die Domäne
- Zur Strukturierung gibt es OUs (Organizational Units)

AD-Konten verwalten

- AD-Konten existieren für
    - Benutzer
    - Computer
- Können in Gruppen zusammengefasst werden
- Jedes Konto hat eine Sicherheitskennung SID
- Zur Verwaltung von OUs können User einer OU Objektverwaltungsrechte eingeräumt werden

Verwaltungstools

- AD Benutzer und Computer
- AD Verwaltungscenter
- Kommandozeile
- PowerShell

Container in einer Domäne

- **BuiltIn**: Spezielle lokale Sicherheitsgruppen (z.B. lokale Admins)
- **Computers**: Alle Computerkonten
- **Domain** Controllers: Alle DCs einer Domäne
- **ForeignSecurityPrincipals**: Container für SIDs einer vertrauten Domäne
- **Program** Data: Ablageort für Programmdaten im AD
- **Users**: Benutzerkonten und Gruppenkonten einer Domäne
- **Eigene**: Eigens erstellte OUs