Windows Richtlinien

Sicherheitstage WS 2006/2007

Hergen Harnisch

harnisch@rrzn.uni-hannover.de

21.11.2006

Einführung:

R|R|Z|N|

Ziel

Einstellungen (für mehrere Benutzer / Computer automatisiert) vorgeben (Service) oder erzwingen (Sicherheit)

Mittel

- Benutzerprofile
- NT4-Systemrichtlinien, Sicherheits- & Gruppenrichtlinien
  - benutzerspezifische
  - computerspezifische

Wirkungsweise

Richtlinien u.a. Übernahme von Einträgen in die Registry
Registry- & Rechteübernahme mit System-Rechten

Profile u.a. Dateien unter `Dokumente und Einstellungen`
Registry- & Dateiübernahme mit User-Rechten

Ausführungen gedacht für 2000, XP, 2003

R|R|Z|N|

Zentrale Windowsdatenbank zu System- & Programmeinstellungen

aktive Anwendungseinst.
akt. angemeldeter Benutzer
inst. Hard- & Software
benutzerspez. Einst. aller B.
Kopie `HKLM` je Hardwareprof.

`HKCR`: nur ein View, entspricht `HKLM\Software\Classes` mit `HKCU\Software\Classes` überlagert

Einführung: Profile vs. Richtlinien

R|R|Z|N|

Profile

- Registry-Einträge nur unterhalb von `HKCU`
- in Dateien `NTUSER.DAT` und `NTUSER.MAN` abgelegt

Richtlinien

- Registry-Einträge unterhalb von
  - `HKLM` als „Computerkonfiguration“
  - `HKCU` als „Benutzerkonfiguration“
- Sicherheitsrichtlinien
- Gruppenrichtlinien im lokalen GPO und in AD-GPOs
- NT4-Systemrichtlinien in `NTCONFIG.POL`-Datei

R|R|Z|N|

aktive Gruppen- und Systemrichtlinien in der Registry:

← Benutzerkonf.

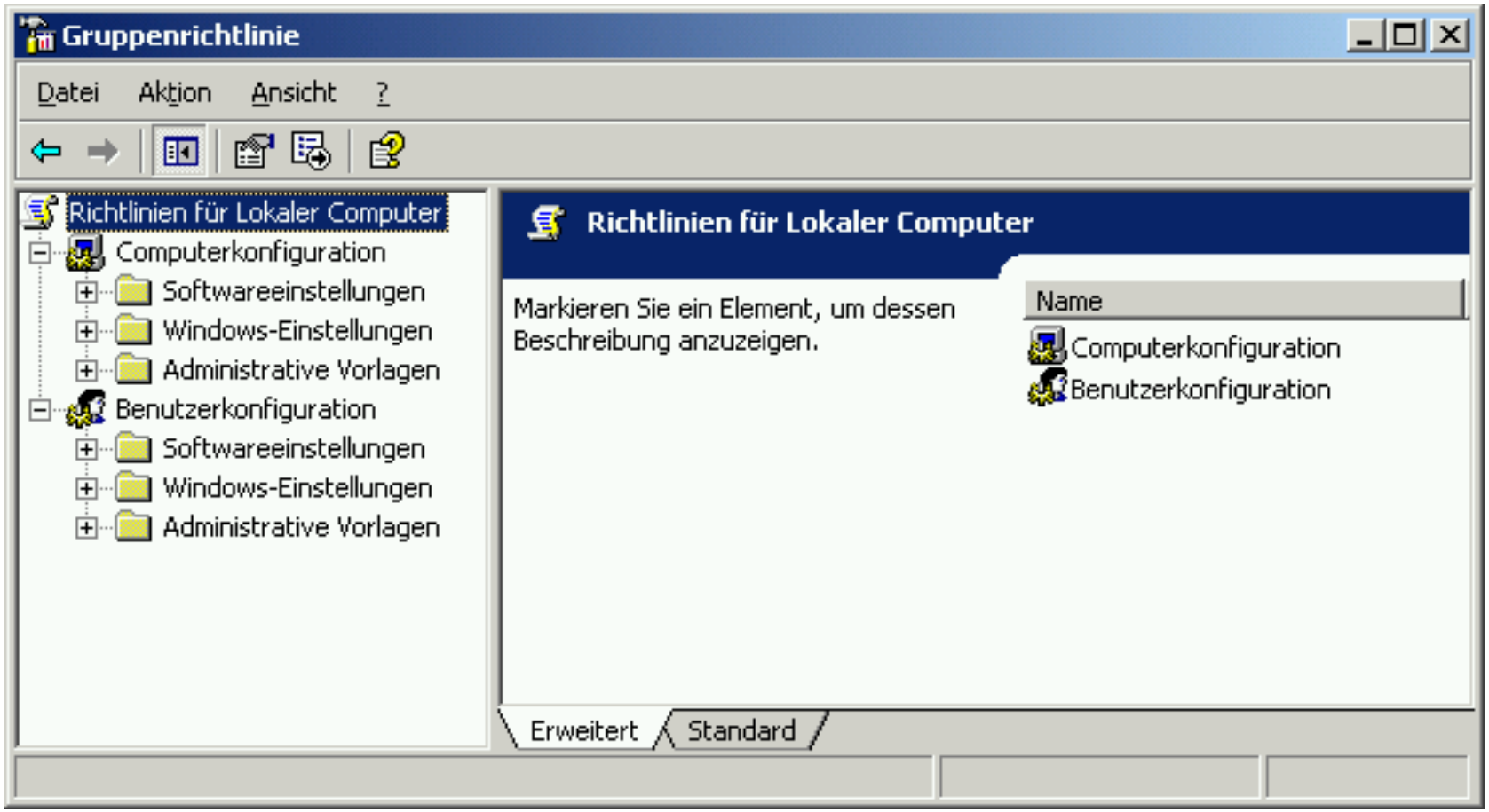
← Computerkonf.

`HKCU`: Richtlinien überschreiben Profil

R|R|Z|N|

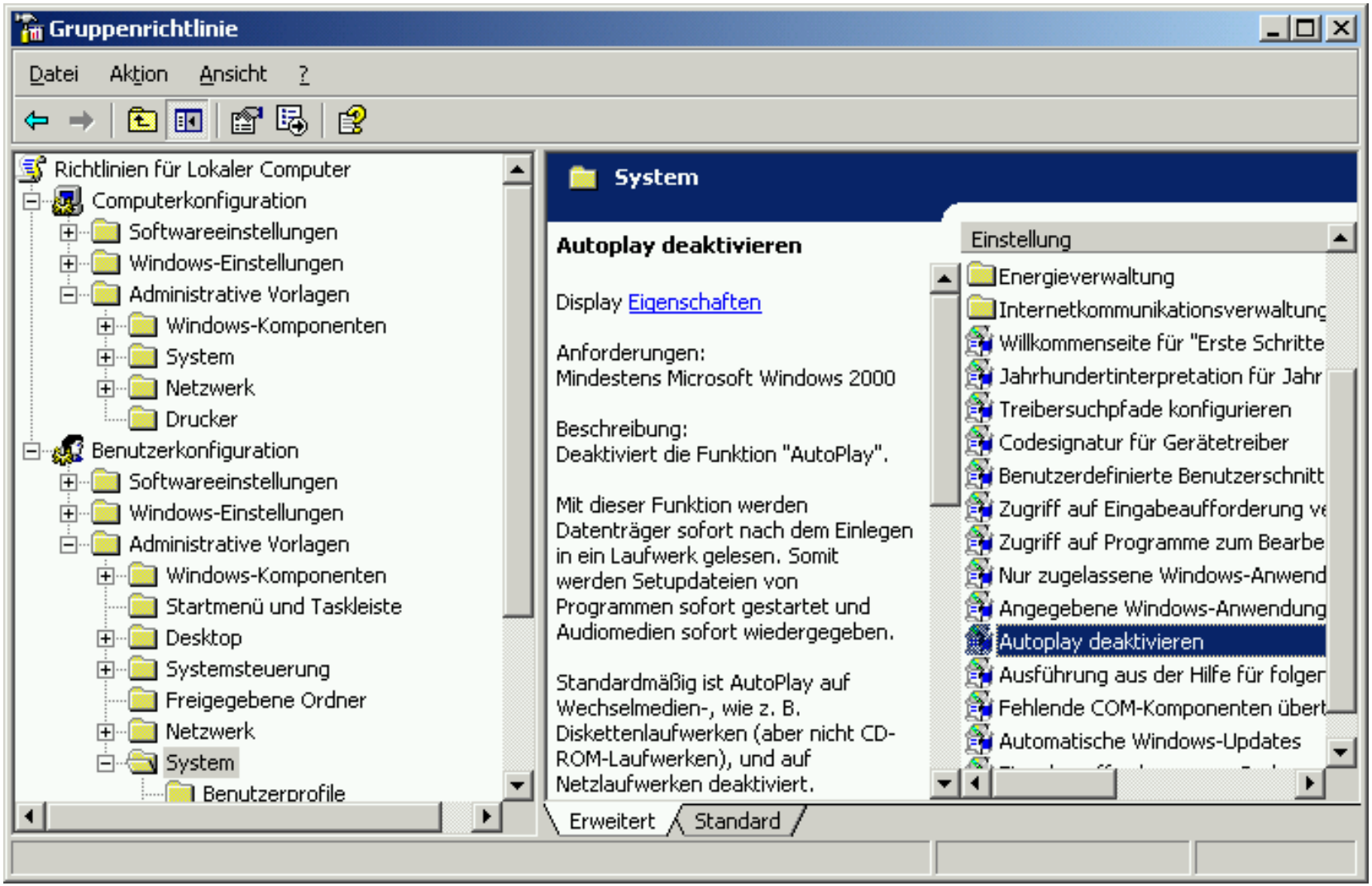
GPO-Bearbeitung (GUI)

mit MMC-Snapin `gpedit.msc` (Gruppenrichtlinie):



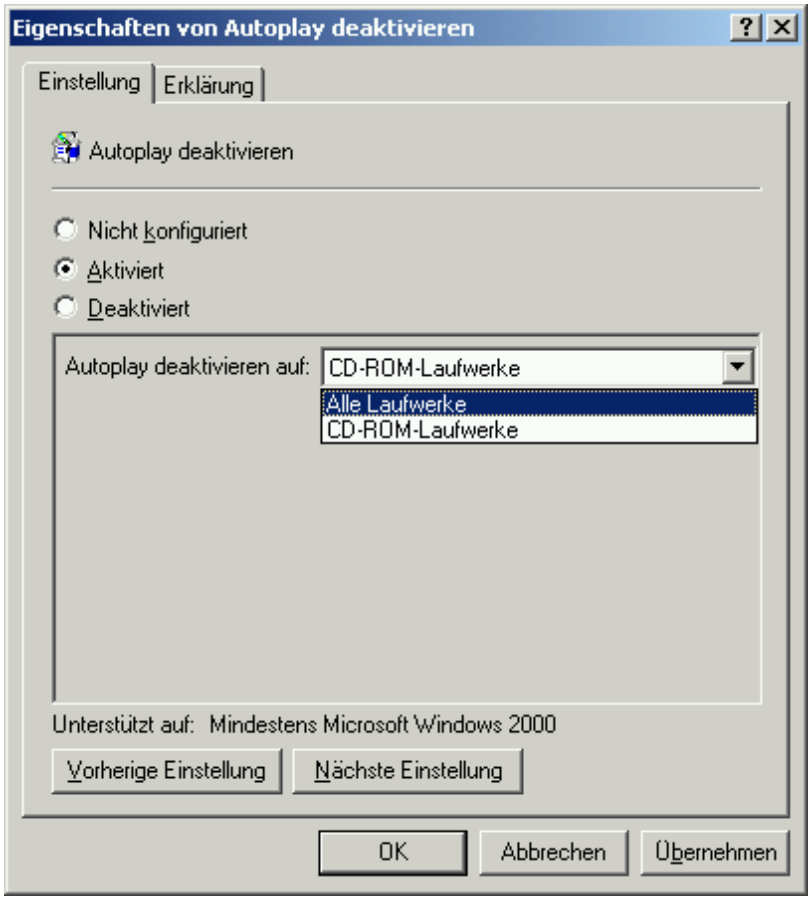
R|R|Z|N

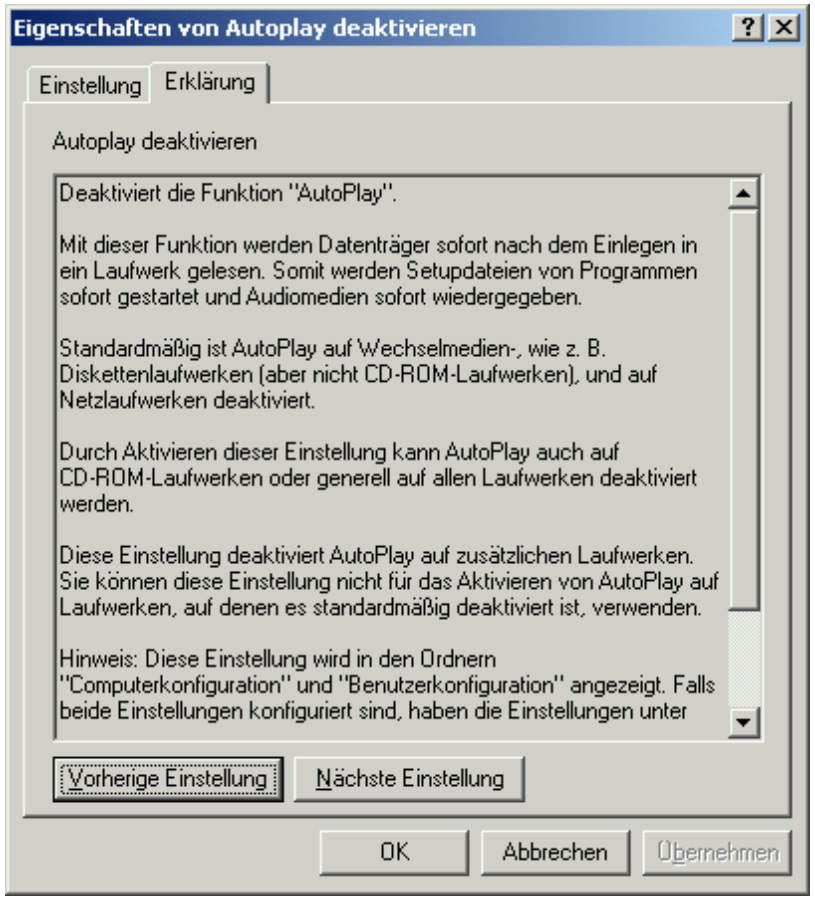
Beispiel: Autoplay-Funktion



R|R|Z|N

Beispiel: Autoplay-Funktion





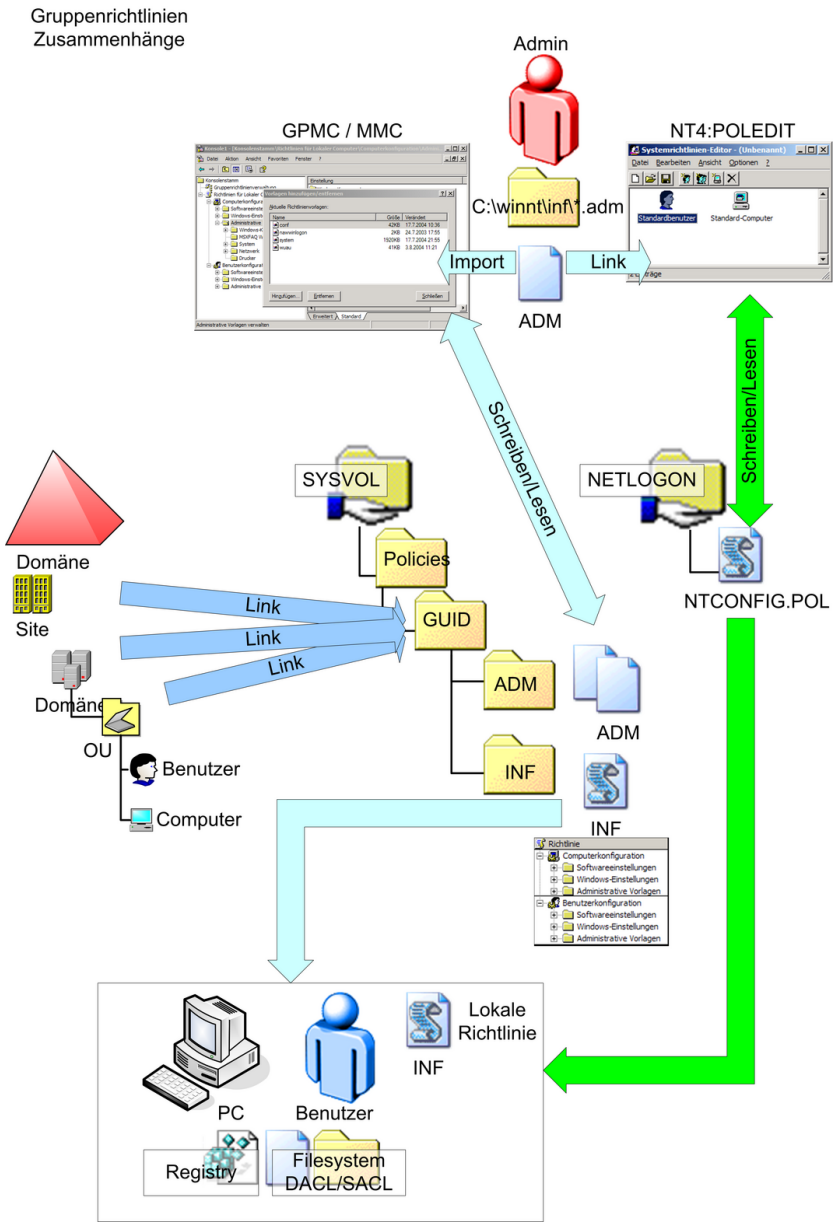
aber: wirksam erst nach `gpupdate /force` oder Neustart

R|R|Z|N|

Ablauf & Priorität

in abnehmender Priorität:

1. NT4-Richtlinien
2. lokales GPO
3. GPOs in Domäne



Quelle:

Frank Carius, MS Exchange FAQ

www.msexchangefaq.de

lokales GPO

R|R|Z|N|

Gilt für den Computer und alle Nutzer:
außer über Dateirechte keine Auswahl möglich (keine Leseberechtigung $\Longrightarrow$ kein Einlesen der Benutzerkonfiguration)

Dateipfad[1]

- `%SYSTEMROOT%\system32\GroupPolicy\Machine\Registry.pol` enthält Computerkonfiguration, Einlesen beim Booten
  $\longrightarrow$ Registry-Zweig `HKEY_LOCAL_MACHINE`
- `%SYSTEMROOT%\system32\GroupPolicy\User\Registry.pol` enthält Benutzerkonfiguration, Einlesen bei Anmeldung
  $\longrightarrow$ Registry-Zweig `HKEY_CURRENT_USER`

Verteilen

Dateien können auf andere Rechner kopiert werden,
Richtlinien aktivieren mit `gpupdate /force` oder Neustart

[1] ohne Sicherheitsrichtlinien

R|R|Z|N|

REGISTRY.POL direkt bearbeiten / skripten

Dateiformat

Das Dateiformat ist (inzwischen) von Microsoft dokumentiert (`http://msdn2.microsoft.com/en-us/library/aa374407.aspx`):

1. DWORD 0x67655250 („PReg“)
2. DWORD 0x1 (Format-Version)
3. in Unicode-Text bzw. binär: [key;value;type;size;data]

Freeware-Tool

`gpscript` bearbeiten des IGPO über Kommandozeile

`gpcvreg` übertragen von `.reg`-Dateien in IGPO

`http://www.mirkes.de/de/freeware/batch.php` (GPL, Delphi)

R|R|Z|N|

gpscript

- Dump: `gpscript` *`Registry.pol`* `/dump >` *`Datei.txt`*
- Restore: `gpscript` *`Registry.pol`* `/file <` *`Datei.txt`*
  fügt zu, ggf. vorher: `gpscript` *`Registry.pol`* `/clear`
- registry-basierte Richtlinien
  - `/key:"..." /value:"..." /type:.../data:...`

  editieren
  - `/add` zufügen, wenn noch nicht vorhanden
  - `/modify` ändern falls bereits vorhanden
  - `/delete` entfernen (so vorhanden)

(statt *`Registry.pol`* ist auch `MACHINE` bzw. `USER` möglich)

R|R|Z|N|

gpscript: Anwendungsbeispiel

1. Nutzer-Richtlinien auf einem Rechner mit `gpedit.msc` setzen
2. `gpscript USER /dump > lgpo-user.txt`
3. ggf. Nachbearbeiten (z.B. nur einige Richtlinien) mit Editor:

```
# ... Kommentare ...
/KEY:"Software\Microsoft\Windows\CurrentVersion\
                                    Policies\Explorer"
/VALUE:"NoDriveTypeAutoRun"
/TYPE:REG_DWORD
/DATA:D 0x000000FF
/SET
```

4. Zielrechner: `gpscript USER /file:lgpo-user.txt`
   z.B. per Startupskript mit `lgpo-user.txt` aus Netlogon-Share
5. lGPO laden: `gpupdate /force`

R|R|Z|N|

`gpscript`: Vorteile

- Skripting gegenüber GUI generell:
  - Dokumentierbarkeit
  - Automatisierung
- Update-Möglichkeit, ggf. Rücksicht auf individuelle lGPOs
- Modularität:
  als Einzeldateien austauschbar, zusammenstellbar
- automatische Anpassung von `gpt.ini` (Versions-Inkrement)

NT4-Richtlinien

R|R|Z|N|

- von Microsoft bezeichnet als „Systemrichtlinien“
  für $\geq$ 2000 von MS dokumentiert (NT4-Domäne, Workgroup, lokal)
- nur bezogen auf Registry-Werte,
  ggf. *tattooing* (kein Rücksetzen bei Richtlinienentfernung)
- Richtlinien je nach User oder Computer möglich
- Grundlage / Policy-Definitionen in `.adm`-Dateien
- abgelegt in Policy-Datei (Endung `.pol`),
  Aktivierung auch bei Neustart oder `gpupdate /force`
- Beachtung muss explizit konfiguriert werden (Registry): [2]
  `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Update`
- Bearbeiten mit `poledit` (Microsoft) oder `editreg` (Samba)

[2] bei Windows 2000, XP, 2003

R|R|Z|N|

Samba-DC

- Policy-Datei liegt auf dem Netlogon-Share auf dem DC als Datei `NTConfig.POL`[3]
- Wird dort automatisch gesucht, wenn

  `HKLM\System\CurrentControlSet\Control\Update\UpdateMode=1`

lokal / Workgroup / Novell

- Registry-Patch ist nötig:

  `HKLM\System\CurrentControlSet\Control\Update\UpdateMode=2`

  `HKLM\System\CurrentControlSet\Control\Update\NetworkPath=`*Pfad*`\`*Datei*
- Dateirechte für Policy-Datei beachten (alle lesen, nur Administrator schreiben), z.B. automatisch bei `%SYSTEMROOT%\System32\GroupPolicy\NTConfig.POL`

[3] bei Windows NT4, 2000, XP, 2003; war bei 98 & ME `Config.Pol`

R|R|Z|N|

Richtlinien-Update auf XP / .adm-Dateien

1. .adm-Dateien aus z.B. admFiles_WindowsXPSP2.msi (Microsoft-Download) oder %SYSTEMROOT%\inf*.adm
2. Entfernen der Anweisungen #if version <= 2 & #endif
3. unbedingt neueres Poledit.exe aus $\geq$ 2000 verwenden

$\longrightarrow$ *fertig angepasst inkl. Poledit von www.gruppenrichtlinien.de/Info/Downloads.htm*

5. .adm-Dateien einlesen mit Poledit.exe über Options/Policy Templates (dauert!)
6. ggf. weitere .adm-Dateien (selbst erstellte oder aus dem Internet, z.B. für MS-Office) hinzufügen

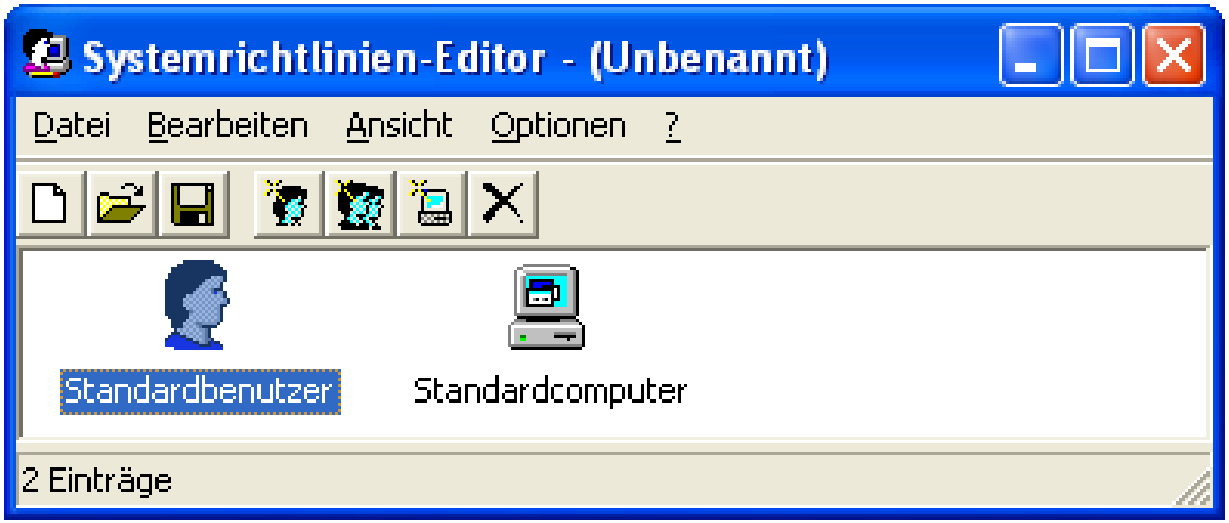
R|R|Z|N|

Poledit: ADM-Vorlagen

- NT4-angepasste `.adm`-Dateien nicht nach `%SYSTEMROOT%\inf\` sondern separat, da sonst auch in GPOs genutzt und bei Updates überschrieben.
- zunächst System-„Richtlinienvorlagen“ in Poledit rauslöschen und NT4-angepasste einlesen (Einlesen dauert)

R|R|Z|N|

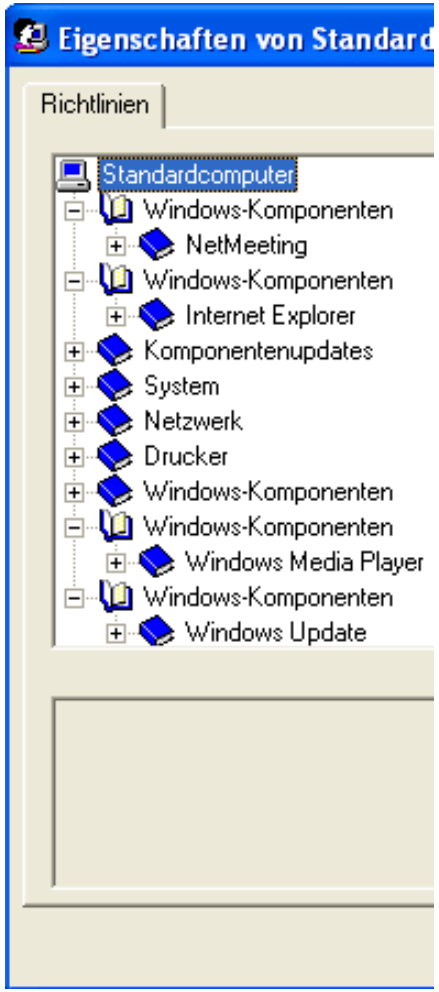
Poledit: Richtlinie



- Datei / „Neue Richtlinie“ startet mit Standardnutzer & -computer, spezielle Nutzer, Computer, Gruppen anlegbar (müssen nicht auf System existieren)
- bearbeitete Richtlinie in `.POL`-Datei speichern, evt. betriebssystem- oder computerabhängige Dateien

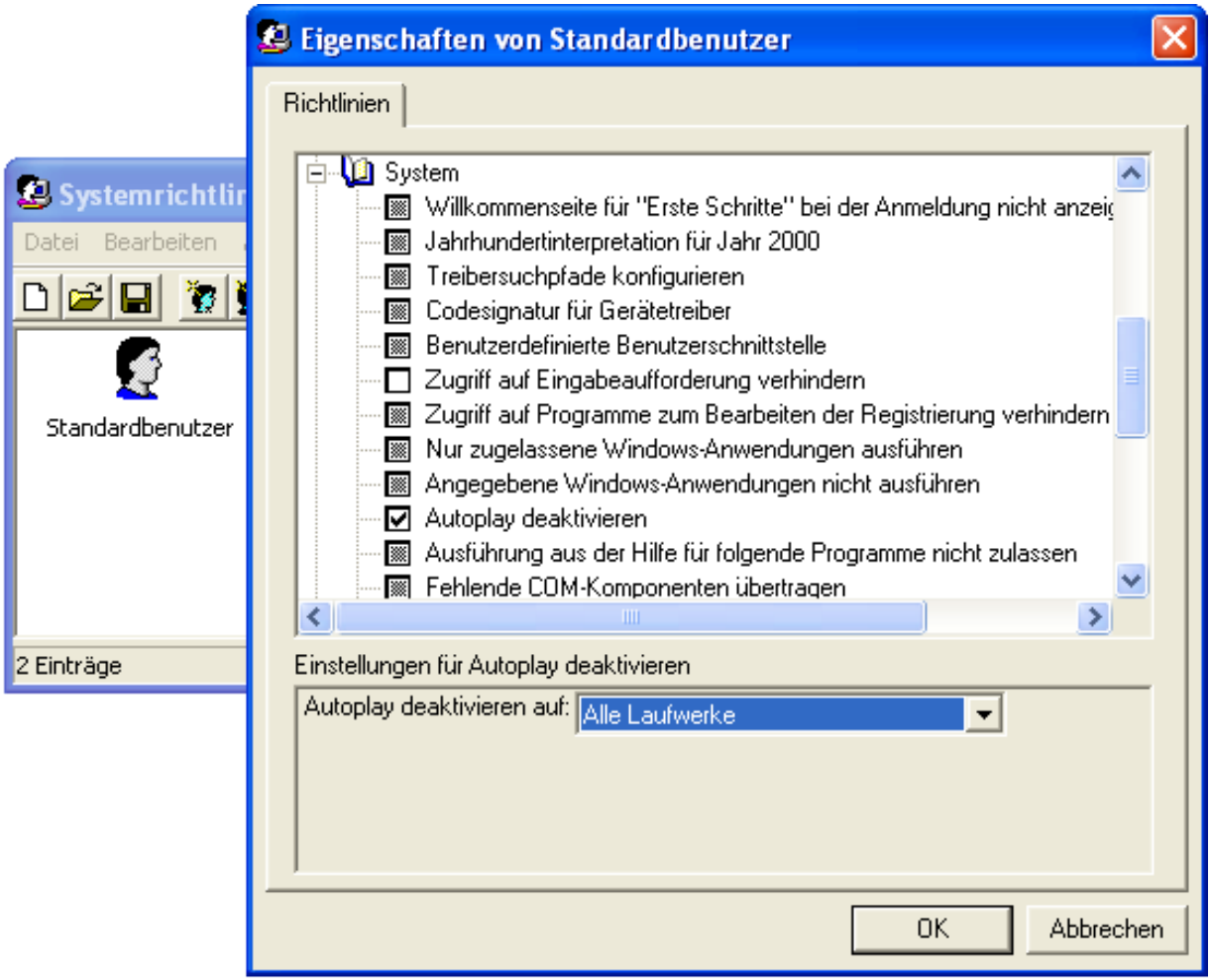
R|R|Z|N|

Poledit: Nutzer & Computer



R|R|Z|N|

Poledit: Richtlinienbearbeitung



R|R|Z|N|

Poledit: Richtlinienbearbeitung

- ▩ Richtlinie nicht gesetzt
  Wert der Registry wird nicht geändert
- ☑ Richtlinie aktiviert, evt. Zusatzwerte
  Werte werden in Registry geschrieben
- ☐ Richtlinie aufgehoben
  entsprechende Werte der Registry werden aufgehoben

Tattooing

Registry-basierte Richtlinien, die einmal gesetzt wurden, bleiben beim Rücksetzen auf „nicht gesetzt“ in der Registry erhalten. Ausnahme: Richtlinien, die in Policy-Ästen der Registry liegen.

R|R|Z|N|

Troubleshooting

Registry

Im Policy-Editor über Datei / „Registrierung öffnen“ ist aktive Systemrichtlinie einsehbar.

Policy-Datei

Die `.POL`-Datei ist im Registry-Editor ladbar (unterhalb eines Schlüssels importieren, dann wieder löschen; sogenannter *hive*)

Sicherheitsrichtlinien

R|R|Z|N|

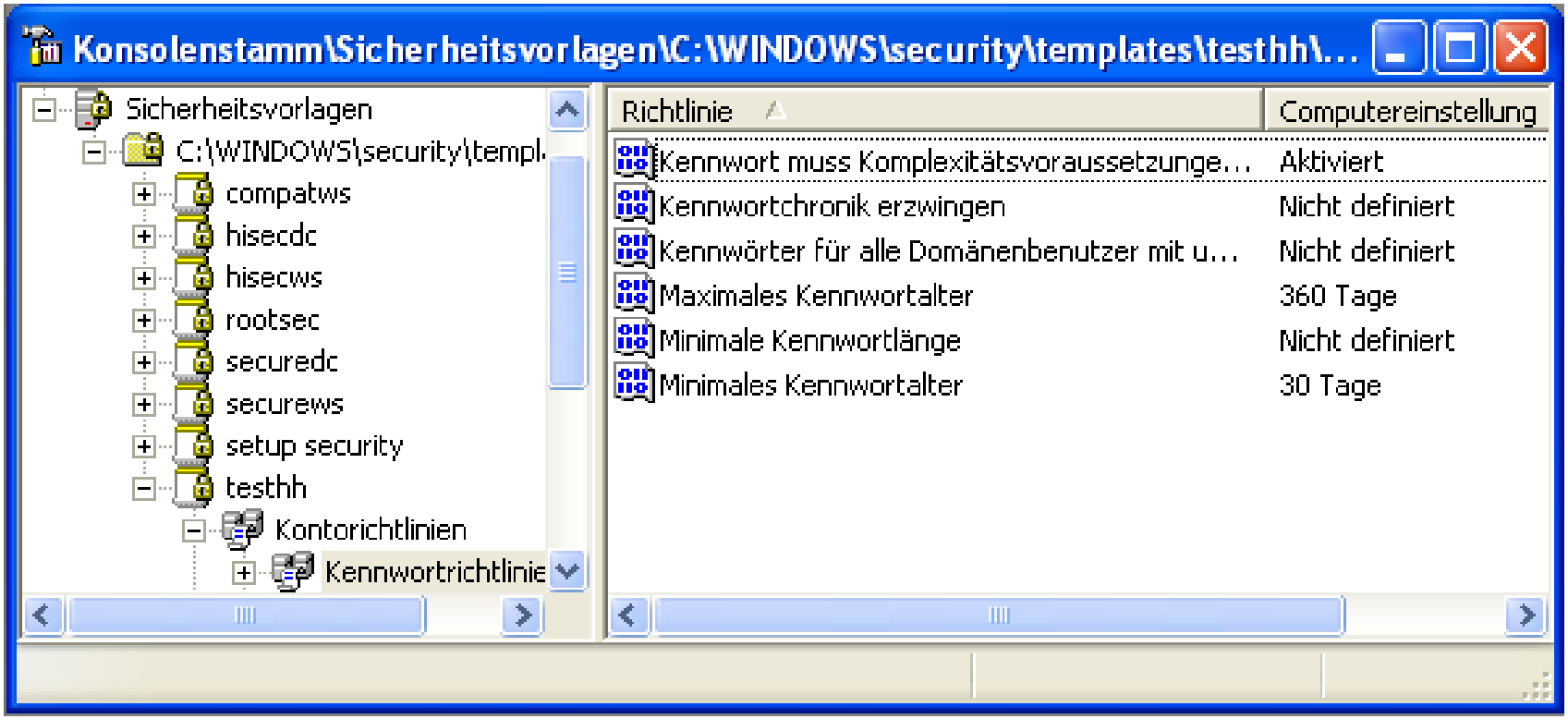
- separates MMC-Snapin `secpol.msc`,
  auch über `gpedit.msc` (unter Windows-Einstellungen)
- lokal *nicht* enhalten in `Registry.Pol`-Dateien
- Automatisierung nur über Templates:
  - mit MMC-Snapins
    - Sicherheitsvorlagen:
      erstellen und bearbeiten
    - Sicherheitskonfiguration & -analyse:
      einlesen und Kontrolle

    und `secedit.exe`
  - in `.inf`-Dateien als SDDL-Text editierbar

R|R|Z|N|

Templatebearbeitung

innerhalb der MMC mit dem Snapin „Sicherheitsvorlagen“



R|R|Z|N|

Template-Datei

Unicode-Text in SDDL-Syntax:

```
[Unicode]
Unicode=yes
[Version]
signature="$CHICAGO$"
Revision=1
[System Access]
MinimumPasswordAge = 30
MaximumPasswordAge = 360
PasswordComplexity = 1
[Registry Values]
```

Einspielen mit

```
secedit /configure
```

R|R|Z|N|

Microsoft-Detours

- Richtlinien liegen in der Registry, werden dort lesend abgefragt.
- Detours erlaubt das Umbiegen von Windows-API-Calls: *„It is a library for intercepting arbitrary Win32 binary functions on x86 machines.“*
- Änderung von `RegQueryValueExW` mit Detours: Anfrage weiterreichen an Windows-Originalfunktion, nur bei zu umgehenden Richtlinien (d.h. Registry-Keys) eigene Werte zurückgeben.
- Programme fragen danach „falsche Richtlinien“ ab.

vgl. `http://www.codeproject.com/system/KamalDetours01.asp`
ähnliches Vorgehen mit `gpdisable` / DLL-Injektion

R|R|Z|N|

lokale Rechte

lokaler Administrator

relevant in einer Domäne mit GPOs:

- Lokaler Administrator kann Registry überschreiben
- damit registry-basierende Policies umgehbar
- Policy-Update kann unterbunden werden

aber:

- nicht angreifbar, was auf dem DC liegt
  z.B. Kennwort-Richtlinien für Domänen-Accounts

elevated rights

ggf. bei der Programminstallation erhöhte Rechte

vgl. Richtlinie „Immer mit erhöhten Rechten installieren“

R|R|Z|N|

Falscher Ansatzpunkt

Richtlinien setzen häufig nicht an der richtigen Stelle an, z.B.:

Registry

- Programmaufruf von `Regedit.exe` wird verboten,
- Registry-Wert ist aber überschreibbar.

Programmausführung

- Internet-Explorer `iexplore.exe` wird verboten,
- Aufruf über Dateiumbennung oder Eingabeaufforderung mögl.

Dateizugriff

- Laufwerke werden ausgeblendet,
- Laufwerk/Datei durch direkte Pfadeingabe aufrufbar.

R|R|Z|N|

Windows-Absicherung

Sicherheit muss an der passenden Stelle ansetzen:

- Dateizugriffe über Dateirechte
- Registry-Daten über Registry-Rechte
- Organisatorisches über Richtlinien, evt. mehr

aber: lokale Exploits kaum auszuschließen!

R|R|Z|N|

Vorteile

Trotz der Schwächen sind Richtlinien sinnvoll:

- einige Einstellungen garantieren Sicherheit
- Abwehr von „Gelegenheitshackern“, Hack wäre evt. Umgehung wirksamer technischer Schutzmaßnahme
- Schutz des Nutzers vor sich selbst (z.B. IE-Einstellungen, Autostart bei CD/USB)
- Vereinfachte Administration
  - sinnvolle Vorgaben für Nutzer nicht einzeln vorzunehmen
  - im Netz zentral verteil- und änderbar
  - evt. weniger Support (z.B. verschobene Taskleiste)
  - erweiterbar durch eigene ADM-Vorlagen oder aus dem Netz, teilweise für Anwendungen (z.B. von Microsoft für Office)

R|R|Z|N|

Deployment ohne AD

- für generelle Vorgaben Gruppenrichtlinien, per `gpscript` einspielen / ändern
- für nutzerabhängige Vorgaben NT4-Systemrichtlinien, per `.POL`-Datei
- für rechnerabhängige Vorgaben
  - skriptgesteuerte `gpscript`-Lösung (mächtiger)
  - oder per `.POL`-Datei (einfacher)

GPO und NT4 `.POL`-Datei schließen sich nicht aus![4]

mit Fileserver:

- `gpscript`-Datei / -Skript auf Server, Startup-Skript auf Client
- `.POL`-Datei auf Share (Pfadangabe in Registry eintragen)

[4] IGPO wird von `.POL`-Datei überlagert.

R|R|Z|N|

Deployment

- Zwischen Einstellungen in Richtlinien und Profilen entscheiden
- nicht zu viele Richtlinien / Details
- Nutzerwünsche berücksichtigen
- Einstellungen dokumentieren

Häufigster Fehler

Aktivierung mittels `gpupdate /force` vergessen.

R|R|Z|N|

Bearbeitung

Vorbereitung

Um die NTFS-Rechte im Windows-Explorer sehen zu können: Deaktivieren von „Einfache Dateifreigabe verwenden“ im Windows-Explorer („Extras“, „Ordneroptionen“, „Ansicht“)!

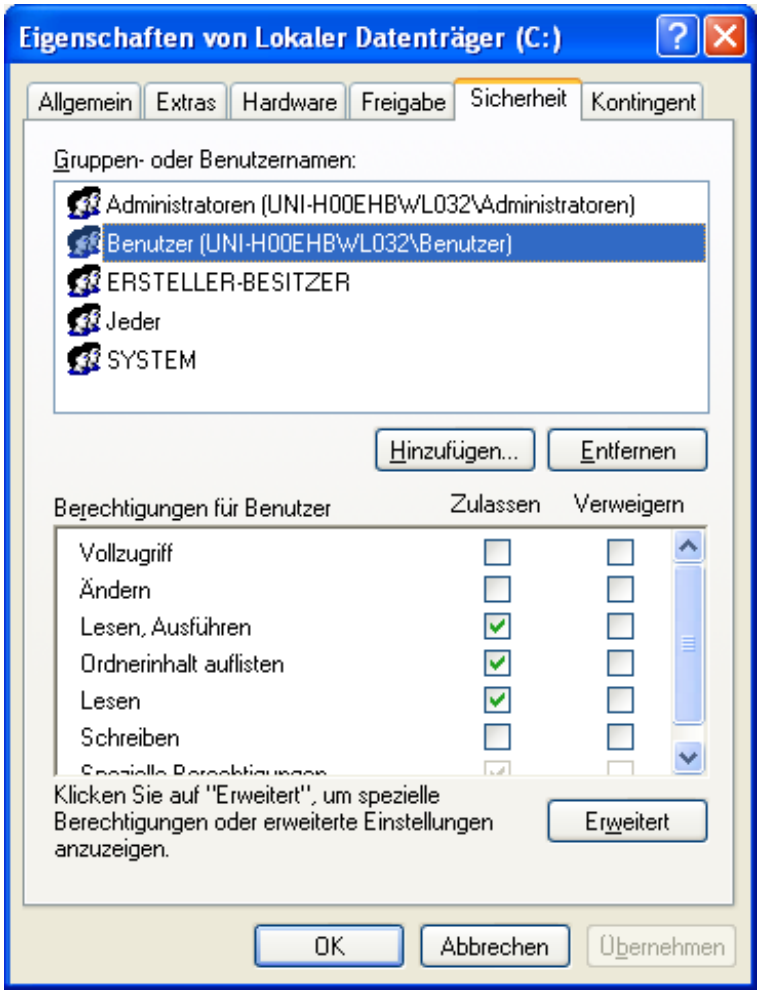
Tools

zum Anzeigen und Ändern:

- Windows-Explorer für Verzeichnisse und Dateien unter „Eigenschaften“, Kartei „Sicherheit“
- Kommandozeilen-Tool `cacls.exe`

R|R|Z|N|

Bearbeitung



- Gruppen und Nutzern
- können Rechte gegeben
- oder explizit genommen werden.
- Vollzugriff, Ändern, Lesen/Ausführen sind Abk.
- Ererbte erscheinen grau,
- hier definierte grün.

Dateirechte

R|R|Z|N|

Rechte

Neben den Üblichen zum Auflisten, Lesen und Schreiben gibt es noch spezielle: Attribute ändern, Anhängen, Ordner erstellen, Berechtigungen ändern ...

Rechteinhaber

Benutzer und Gruppen können Rechte haben, zudem noch das System und der „Ersteller-Besitzer“.

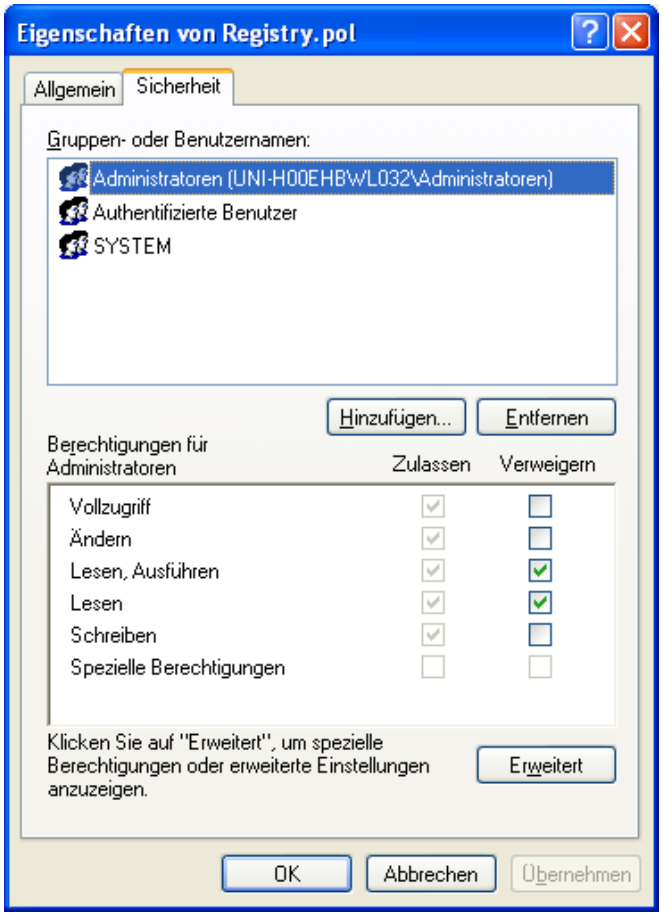
Gültigkeit

Rechte nach unten im Dateibaum vererbbar (Standard), nur aktuelles Verzeichnis, nur in Unterordnern (sowie Kombinationen).

R|R|Z|N|

Zugriff verhindern

- durch Gruppen oder übergeordnete Verzeichnisse geerbte Rechte nehmen
- z.B. Benutzer-IGPO nicht für Administrator gelten lassen
  Achtung: `gpedit.msc` / `gpscript.exe` können evt. auch nicht lesen



R|R|Z|N|

spezielle/erweiterte

Registryrechte

R|R|Z|N|

Berechtigungen gibt es nur für ganze Schlüssel, nicht für einzelne Werte.

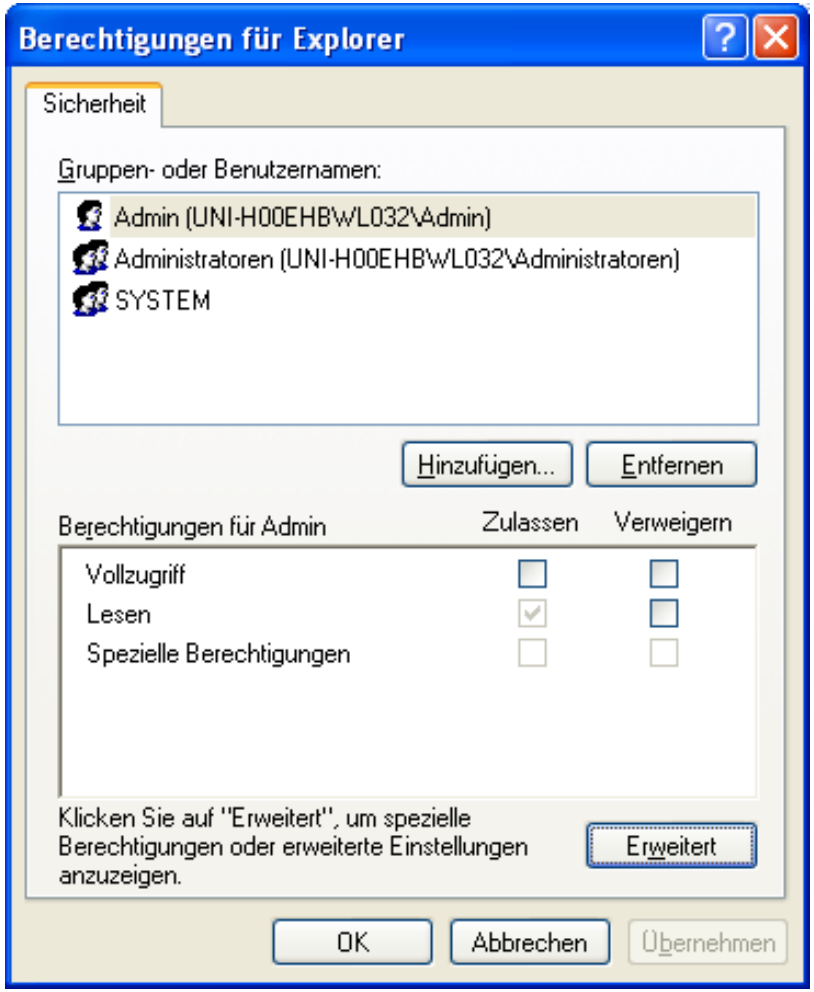
Tools

zum Anzeigen und Ändern:

- Registry-Editor zeigt unter Bearbeiten/Berechtigungen die Rechte an.
- Kommandozeilen-Tools:
  - `regini.exe`: kryptisch, Ändern von Rechten & Werten
  - GPL: `setacl.exe`: Ändern, Backup/Restore von Rechten, auch für Dateien & Verzeichnisse, Drucker, AD-Objekte ...

R|R|Z|N|

Bearbeitung



- Gruppen und Nutzern
- können Rechte gegeben
- oder explizit genommen werden.
- Ererbte erscheinen grau,
- hier definierte grün.

Links

R|R|Z|N|

- XP und (L)GPO, Systemrichtlinien etc.:
  `http://www.microsoft.com/technet/prodtechnol/winxppro/reskit/`
- `gpscript.exe`:
  `http://www.mirkes.de/de/freeware/batch.php`
- Systemrichtlinien mit Samba:
  - `http://samba.org/samba/docs/man/Samba-HOWTO-Collection/PolicyMgmt.html`
  - `http://www.pcc-services.com/custom_poledit.html`
- Policy-Editor und dafür angepasste `.adm`-Dateien:
  `http://www.gruppenrichtlinien.de/Info/Downloads.htm`